UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

TRABAJO FIN DE MÁSTER

DESARROLLO DE UN SISTEMA DE TRAZABILIDAD EN ENTORNOS IOT MEDIANTE HYPERLEDGER

Máster Universitario en Ingeniería Informática

AUTOR: IGLESIAS GARCÍA, JESÚS

TUTOR: ARROYO GUARDEÑO, DAVID

SEPTIEMBRE 2018

UNIVERSIDAD AUTÓNOMA DE MADRID
Escuela Politécnica Superior
Francisco Tomás y Valiente, nº 11
Cantoblanco, Madrid, 28049
España

Desarrollo de un sistema de trazabilidad en entornos IoT mediante Hyperledger
Jesús Iglesias García, Septiembre 2018

IMPRESO EN ESPAÑA - PRINTED IN SPAIN

DESARROLLO DE UN SISTEMA DE TRAZABILIDAD EN ENTORNOS IOT MEDIANTE HYPERLEDGER

AUTOR: IGLESIAS GARCÍA, JESÚS
TUTOR: ARROYO GUARDEÑO, DAVID

DEPARTAMENTO DE INGENIERÍA INFORMÁTICA
ESCUELA POLITÉCNICA SUPERIOR
UNIVERSIDAD AUTÓNOMA DE MADRID
SEPTIEMBRE 2018

Agradecimientos

Dedicado a mis padres por el gran esfuerzo que han hecho durante toda mi vida para apoyarme durante la consecución de todas las metas propuestas tanto personales como profesionales.

Agradecer en especial a David Arroyo Guardeño, tutor de este Trabajo Fin de Máster, por su gran dedicación y apoyo recibido de forma desinteresada tanto en este trabajo como durante mi paso por la Escuela Politécnica Superior. Sin duda alguna, un gran profesor y mejor persona.

Mención también a la Escuela Politécnica Superior por la ayuda proporcionada para la obtención del hardware empleado en el Trabajo Fin de Máster.

Resumen

En la actualidad un grueso importante de empresas tiene un interés especial en construir una nueva generación de aplicaciones transaccionales que establezcan confianza, responsabilidad y transparencia en su núcleo junto con una arquitectura abierta y distribuida. Aquí, es de especial relevancia el sistema de pago de gestión no centralizado Bitcoin, que está basado en la tecnología Blockchain (BC) la cual habilita la creación de un sistema inmutable de registro de sucesos. Desde su origen ha propiciado todo un conjunto de iniciativas que, prescindiendo de la existencia de una tercera parte de confianza (TPC), puedan proporcionar protocolos para la protección de la integridad de la información.

Una de estas iniciativas es Hyperledger -estándar *open source* de investigación y evolución de la tecnología BC permisionada anunciada por la fundación Linux en 2015-, se centra en apoyar este tipo de transacciones de negocio para mejorar numerosos aspectos de rendimiento y empresariales. Hyperledger Fabric (HF), es el proyecto más conocido dentro de este consorcio. Implementa la tecnología de libro de registros distribuido (*Distributed Ledger Technology* -DLT-) en una BC permisionada donde la inserción de la información requiere que las entidades estén previamente autenticadas. Este modelo de control de acceso es de alto interés en el contexto de la trazabilidad de recursos ya que limita quién puede tener acceso y a qué recursos gracias a la existencia de roles.

El surgimiento de nuevos paradigmas, como el Internet de las Cosas (*Internet of Things* -IoT-) donde cualquier objeto del mundo real puede estar totalmente conectado a la red, ha evolucionado la forma de uso, de comunicación y de transmisión de información a través de Internet. Hasta hace poco, estos dispositivos eran meros consumidores de datos. Actualmente, exponen de forma permanente servicios y datos recolectados del entorno con lo que aparte de traer consigo numerosos beneficios, también abren la puerta a numerosos problemas de seguridad y privacidad. La importancia de mantener un registro de información inmutable y confidencial es esencial para garantizar la veracidad e integridad de los sucesos y de la información recogida. El presente proyecto aborda el desarrollo de una prueba de concepto (*Proof of Concept* -PoC-) para implementar HF en el IoT. El objetivo es la recolección de eventos mediante sensores situados en una Raspberry Pi 3 (RPi), y la inclusión de incidencias en la BC de HF. Asimismo, toda la información recolectada puede ser consumida en tiempo real mediante un sistema web.

Palabras clave

Cadena de bloques, Contratos inteligentes, Hyperledger, Hyperledger Fabric, Internet de las Cosas, Libro principal, Privacidad, Raspberry Pi, Seguridad, Sensor

Abstract

Currently, a significant number of companies have a special interest in developing a new generation of transactional applications which establish trust, responsibility and transparency in their core, together with an open and distributed architecture. Of particular relevance here it is the non-centralized payment system Bitcoin, which is based on the Blockchain (BC) technology that enables the creation of an immutable system of event registration. Since its inception, it has led to a whole set of initiatives that, regardless of the existence of a trusted third party (TPC), can provide protocols for the protection of the integrity of information.

One of these initiatives is Hyperledger -open source standard of research and evolution of the permissioned BC technology announced by the Linux Foundation in 2015-, focuses on supporting this type of business transactions as a means to improve numerous aspects of the technology performance and the business activities. Hyperledger Fabric (HF), is the most well-known project within this consortium. It implements the Distributed Ledger Technology (DLT) in a permissioned BC where inserting information requires entities to be previously authenticated. This model of access control is of high interest in the context of asset tracking and traceability, since it limits who can access an asset and which operations are allowed.

The emergence of new paradigms, such as the Internet of Things (IoT) where any real-world object can be totally connected to the network, have modified the way information is used, exchanged, and transmitted through the Internet. Until recently, these devices were mere data consumers. Currently, they provide services and data that is collected continuously from their environment, which apart from bringing numerous benefits, also open the door to numerous security and privacy problems. The importance of maintaining a record of immutable and confidential information is essential to guarantee the veracity and integrity of the events and the information collected in such an environments. This project addresses the development of a Proof of Concept (PoC) to implement HF in the IoT. The purpose is the gathering of events using sensors located on a Raspberry Pi 3 (RPi), and the inclusion of incidents in the BC of HF. Likewise, the information collected may be consumed in real time through a web system.

Keywords

Blockchain, Hyperledger, Hyperledger Fabric, Internet of Things, Ledger, Raspberry Pi, Security, Sensor, Smart Contracts, Privacy

Índice general

Índice de figuras

Índice de tablas

Glosario

Advanced Encryption Standard (AES) Estándar de encriptación avanzada; también conocido como Rijndael, es un esquema de cifrado por bloques adoptado como un estándar de cifrado por el gobierno de los Estados Unidos. Fue anunciado por el Instituto Nacional de Estándares y Tecnología (*National Institute of Standards and Technology* -NIST-) y es considerado como uno de los algoritmos más populares usados en criptografía simétrica. XLIII, 266

American Standard Code for Information Interchange (ASCII) Código Estándar Estadounidense para el Intercambio de Información; código de caracteres basado en el alfabeto latino. Fue creado en 1963 por el Instituto Estadounidense de Estándares Nacionales (*American National Standards Institute* -ANSI-) como una evolución de los conjuntos de códigos utilizados entonces en telegrafía. XLIII

Android Sistema operativo, basado en GNU/Linux, que se emplea principalmente en dispositivos móviles, aunque actualmente su uso se ha extendido a otro tipo de dispositivos: televisores, automóviles, etc. Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y más tarde, en 2005, compró. 218

Application Programming Interface (API) Interfaz de programación de aplicaciones; conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro *software* como una capa de abstracción. XLIII, 137, 241, 295

Asset Activo; concepto relativo a Hyperledger Composer (HC) y la Blockchain (BC) de Hyperledger Fabric (HF) que hace referencia a los bienes, servicios o propiedades tanto tangibles como intangibles que son almacenados en registros en la Blockchain (BC). Dicho de otra forma, es cualquier cosa del mundo real que pueda ser representado y utilizado en una red de negocio. Este concepto es común a otros tipos de Blockchain (BC). 140, 148

Bitcoin Red P2P (*Peer-to-Peer*) descentralizada y distribuida desarrollada por Satoshi Nakamoto (aunque su identidad real es desconocida) en 2008 que se utiliza como sistema de pago para la transferencia de valor empleando para ello la criptomoneda digital, bitcoin. Se sustenta en la Blockchain (BC) y fue la primera aplicación de ella. V, VII, 1, 2, 13, 14, 16, 17, 204

Blockchain (BC) Cadena de bloques; libro de contabilidad (*ledger*) o base de datos (BBDD) distribuida y descentralizada que es administrada y compartida entre muchas partes diferentes pertenecientes a una red de punto a punto (*peer-to-peer* -P2P-). Registra bloques de información de forma permanente y garantiza la integridad de esta información. V, VII, XLIII, 1, 5, 7, 31, 105, 109, 131, 176, 203, 204, 225, 261, 275

Bourne Shell (sh) Intérprete de comandos considerado con la *shell* original de Unix. Fue desarrollada por Stephen Bourne de la compañía AT&T y liberada en el año 1979 en la versión 7 de Unix. XLV, 138

Bourne-again Shell (Bash) Programa informático cuya función es interpretar por consola comandos e instrucciones de un lenguaje de programación. Está basado en la *shell* de Unix y es el intérprete de comandos por defecto en la mayoría de las distribuciones de Linux. XLIII, 54, 147, 218, 259, 275

Breadboard Placa de pruebas; dispositivo sin soldadura para prototipos temporales con diseños de circuitos electrónicos. Está compuesto de un tablero con orificios que se encuentran conectados eléctricamente entre sí de manera interna, habitualmente siguiendo patrones de líneas, en el cual se pueden insertar componentes electrónicos y cables para su conexión. 218, 234, 250

Bug Error de *software*; problema en un programa informático que desencadena un resultado indeseado. 243

Camera Serial Interface (CSI) Interfaz serie para cámaras; especificación de la MIPI Alliance que define la interfaz entre una cámara digital y un procesador anfitrión. XLIII, 250

Cascading Style Sheets (CSS) Hojas de estilo en cascada; lenguaje de diseño utilizado para definir y crear la presentación de documentos estructurados y escritos con un lenguaje de marcado, como HTML (*HyperText Markup Language*). XLIII, 54, 143

Certificado Autofirmado Fichero informático que asocia una serie de datos de identidad a una persona física, organismo o empresa confirmando de esta manera la identidad digital en Internet. Este tipo de certificado no es firmado por una autoridad certificadora (*Certificate Authority* -CA-) sino que es firmado con la propia clave privada y es únicamente utilizado para desarrollo y pruebas del sistema. XV, 153, 264, 265, 269, 271, 297

Certificado SSL Certificado expedido por una autoridad certificadora (*Certificate Authority* -CA-) que acredita la identidad y las credenciales del servidor de tal manera que se garantiza que es auténtico, real y confiable para los usuarios visitantes proporcionando por tanto seguridad a éstos. 152, 265, 267

Certificate Authority (CA) Autoridad de certificación; entidad de confianza responsable de emitir y revocar los certificados digitales o certificados utilizados en la firma electrónica, para lo cual se emplea criptografía de clave pública. XLIII, 153

Certificate Signing Request (CSR) Solicitud de firma de certificado; bloque de texto cifrado que contiene toda la información de la petición de certificado (nombre, dirección, dominio para el que es generado, clave pública, etc.), la cual será incluida finalmente en el certificado SSL. La autoridad certificadora (*Certificate Authority* -CA-) utilizará esta solicitud para generar el certificado SSL. XLIII, 267

Chaincode Contrato inteligente; Programa informático, almacenado en la Blockchain (BC), el cual nadie controla y todo el mundo puede confiar que es capaz de ejecutarse y hacerse cumplir por sí mismo, de manera autónoma y automática, sin intermediarios ni mediadores como si de un contrato con cláusulas contractuales se tratara. 2, 3, 21, 22, 24

Cloud Foundry Plataforma como servicio (*Platform as a Service* -PaaS-) de código abierto que apoya el ciclo de vida completo de aplicaciones de factor 12 -aplicaciones diseñadas para operar de forma adecuada en la nube cumpliendo 12 requisitos- desarrolladas en diversos lenguajes. 105, 137

Cloudant NoSQL DB Producto *software* de base de datos (BBDD) distribuida en la nube perteneciente al servicio IBM Cloud. Es de tipo no relacional (NoSQL) basada en documentos JSON (*JavaScript Object Notation*) que está completamente gestionada y optimizada para la disponibilidad de datos, durabilidad y movilidad mediante una indexación avanzada. Se respalda en el gestor de bases de datos de código abierto Apache CouchDB. 137, 144, 146, 260, 265, 293

Cookie Galleta informática; archivo creado por un sitio web que contiene una cantidad reducida de información y que se envía entre un emisor y un receptor con el propósito de identificar al usuario almacenando su historial de actividad, de manera que se le pueda ofrecer el contenido más apropiado. 76, 77

Criptografía Asimétrica También conocida como criptografía de clave pública o criptografía de dos claves es un método criptográfico donde se utiliza un par de claves para la comunicación. Por un lado, se encuentra la clave pública que se puede difundir sin ningún problema y por otro lado la clave privada que nunca debe ser revelada. Para enviar un mensaje, el remitente usa la clave pública del destinatario para cifrar el mensaje. Una vez que lo ha cifrado, solamente con la clave privada del destinatario se puede descifrar, ni siquiera el que ha cifrado el mensaje puede volver a descifrarlo. Por ello, se puede dar a conocer perfectamente la clave pública para que todo aquel que se quiera comunicar con el destinatario lo pueda hacer. Este tipo de criptografía es más segura que la criptografía simétrica pero requiere de mayor tiempo. 10, 265

Criptografía Simétrica También conocida como criptografía de clave secreta o privada o criptografía de una clave es un método criptográfico donde se utiliza la misma clave tanto para cifrar como para descifrar mensajes. La seguridad reside en la propia clave privada, y por tanto el principal problema es la distribución de claves entre el emisor y receptor ya que ambos deben usar la misma clave de forma que se tiene que buscar también un canal de comunicación que sea seguro para el intercambio de la clave ya que si una persona no autorizada se hace con dicha clave la comunicación ya no podría considerarse como segura. 266

Código QR Código de respuesta rápida; código de barras bidimensional cuadrada que puede almacenar información codificada. 32, 173, 204, 260, 291

Database as a Service (DBaaS) Base de datos como servicio; modelo de servicio de computación en la nube que permite a los usuarios provisionar, gestionar, configurar y operar con bases de datos (BBDD) y a las aplicaciones acceder a la información, abstrayendo de la necesidad de establecer una configuración de *software* o física de *hardware*. XLIII, 109, 137

Diagrama de Gantt Herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado. 226, 227

Distributed Ledger Technology (DLT) Tecnología de registro distribuido; sistema digital para el almacenamiento en múltiples lugares y al mismo tiempo de transacciones de activos evitando así la necesidad de existencia de una autoridad central que almacene y administre los datos. El tipo de registro distribuido más conocido es Blockchain (BC). V, VII, XLIII, 8

Divisor de Voltaje Sistema de resistencias, habitualmente configurado con dos (R1 y R2), conectadas en serie donde una de ellas debe tener el doble de resistencia eléctrica que la otra. Este sistema tiene como cometido fijar la tensión a un nivel intermedio entre el nivel de alimentación del conjunto (por ejemplo, 5V) y el nivel de tierra. 250

ECMAScript Especificación de lenguaje de programación publicada por ECMA International y actualmente aceptado como el estándar ISO 16262. Comenzó a ser desarrollado en el año 1996, basándose en el popular lenguaje JavaScript (JS) propuesto por la compañía Netscape Communications Corporation. 138

Entropia Aleatoriedad recogida por un sistema operativo (SO) o una aplicación para su uso en criptografía o para otros usos que requieren datos aleatorios. Se suele obtener a partir de fuente de *hardware*, tales como los movimientos del ratón o pulsaciones de teclado. 259

ESLint Proyecto *open source* creado originalmente por Nicholas C. Zakas en 2013 con la finalidad de ofrecer un *linter* para JavaScript. 145

Ethereum Plataforma *open source* y descentralizada propuesta por Vitalik Buterin que permite la creación de acuerdos de contratos inteligentes entre pares, basada en el modelo de Blockchain (BC). También, provee una criptomoneda llamada Ether. 2, 13, 14, 17, 18, 25

Ethernet Estándar, también conocido como IEEE 802.3, de redes de área local (*Local Area Network* -LAN-) que determina las particularidades físicas y eléctricas que debe poseer una red tendida con este sistema. 250

Framework Esquema estandarizado de conceptos, prácticas y criterios que se postulan como base para el desarrollo y/o la implementación de una aplicación. 19, 20, 26, 139, 143, 241, 258, 310

Front-End Parte del *software* que interactúa con los usuarios finales, es decir, se corresponde con la interfaz web que visualizan. 138

Función Hash Función resumen; algoritmo para crear a partir de una entrada, una salida alfanumérica de longitud fija que representa un resumen de toda la información que se le ha proporcionado. 8, 11, 133, 260, 267

General Purpose Input Output (GPIO) Entrada/salida de propósito general; pin o conexión genérica en un chip cuyo comportamiento, ya sea un pin de entrada o de salida, puede ser controlado por el usuario en tiempo de ejecución. XLIII, 218, 250, 252, 260, 286

Git Sistema de control de versiones distribuido y eficiente diseñado por Linus Torvalds que permite llevar un registro de los cambios aplicados a ficheros de un proyecto y coordinar el trabajo entre desarrolladores. 142

GNU Privacy Guard (GPG) Herramienta multiplataforma de cifrado y firmas digitales desarrollado por Werner Koch, que viene a ser un reemplazo de PGP (*Pretty Good Privacy*) pero con la principal diferencia que es *software* libre. XLIII, 31, 204, 259, 291

GNU/Linux Conocido como Linux, es un sistema operativo libre, multiplataforma, multiusuario y multitarea basado en Unix. Su origen se remonta a la combinación de varios proyectos, entre los que destaca GNU y el núcleo Linux, encabezado por Linus Torvalds. 56, 70, 72, 138, 146, 160, 277, 288

Grails Conocido Groovy and Rails, es un *framework* de desarrollo de aplicaciones web desarrollado sobre el lenguaje de programación Groovy que potencia el desarrollo ágil. 54, 143, 265, 310

Groovy Lenguaje de programación e incluso de *scripting* dinámico y orientado a objetos nacido en el año 2003 para potenciar Java proporcionando mayor productividad y flexibilidad gracias a todas las funcionalidad que ofrece. Al ejecutarse sobre la máquina virtual de Java (JVM),

presenta el beneficio de poder acceder directamente a todas las API existentes en Java y por tanto usarse directamente en cualquier aplicación de este lenguaje. 54, 109, 139, 143, 144

Groovy Server Pages (GSP) Tecnología de renderizado de la vista del lado de servidor basado en Groovy. Se considera una versión simplificada de la tecnología JSP (*JavaServer Pages*) pero más poderosa e intuitiva que permite emplear una serie de etiquetas para ofrecer una forma más elegante de programación. XLIII, 144

Ground (GND) Conexión a tierra o masa; terminal de referencia para todas las señales o una ruta común en un circuito eléctrico desde donde se pueden medir todos los voltajes y al que se debe aplicar un voltaje de referencia de cero voltios (0V). XLIII, 250

Hibernate *Framework* de código abierto para la plataforma Java que permite el manejo de bases de datos (BBDD) con el paradigma de programación orientada a objetos (POO) a través de una herramienta de mapeo objeto-relacional (ORM). Con ello, a nivel de la aplicación, las tablas se transforman en objetos, los registros se convierten en atributos del objeto, las claves foráneas se transforman en asociaciones entre objetos y las consultas se traducen en llamadas a métodos. Además, proporciona su propio lenguaje de consultas denominado HQL (*Hibernate Query Language*). 109

Historian Concepto relativo a Hyperledger Composer (HC) y la Blockchain (BC) de Hyperledger Fabric (HF) que hace referencia al registro especializado que almacena información sobre todas las transacciones completadas exitosamente, incluyendo los participantes (*participants*) e identidades (*identities*) que las subieron. 148

Hyperledger Iniciativa de carácter colaborativo anunciado en el año 2015 por la fundación Linux para investigar y evolucionar la tecnología Blockchain (BC) de uso privado y orientada al ámbito empresarial. V, VII, 3, 18, 84, 100

Hyperledger Composer (HC) Herramienta de código abierto, perteneciente a la iniciativa Hyperledger, para desarrollar redes de negocio (*Business Network Definition* -BND-) de Blockchain (BC) de soluciones construidas sobre la infraestructura de Hyperledger Fabric, abstrayendo así la complejidad del desarrollo directo en esta última tecnología. XLIV, 3, 21, 110, 137, 176, 204, 217, 261, 265, 273, 295

Hyperledger Fabric (HF) Proyecto ubicado en la iniciativa Hyperledger que implementa la tecnología de libro de registros distribuido (*Distributed Ledger Technology* -DLT-) en una Blockchain (BC) de tipo permisionada. Ofrece características (arquitectura modular y escalable, red transaccional de alto rendimiento permisionada, privacidad e identidad, contratos inteligentes, etc.) que tienden a mejorar aspectos de productividad y fiabilidad distinguiéndola de otras alternativas. V, VII, XLIV, 3, 19, 55, 105, 131, 176, 217, 225, 261, 274, 275

HyperText Markup Language (HTML) Lenguaje de marcado de hipertexto para el desarrollo y representación visual de páginas web. XLIV, 54, 146

Hypertext Transfer Protocol (HTTP) Protocolo de transferencia de hipertexto; protocolo sin estado de comunicación que permite la transferencia de información en la *World Wide Web*. Define la sintaxis y semántica que utilizan los elementos de *software* de la arquitectura web (clientes, servidores) para comunicarse. XLIV, 56, 105, 148, 191, 260

Hypertext Transfer Protocol Secure (HTTPS) Protocolo seguro de transferencia de hipertexto; protocolo de aplicación basado en el protocolo HTTP (*Hypertext Transfer Protocol*), destinado a la transferencia segura de datos. Proporciona una comunicación segura entre origen y destino. XLIV, 105, 148, 191, 258, 260, 297

Identity Identidad; concepto relativo a Hyperledger Composer (HC) y la Blockchain (BC) de Hyperledger Fabric (HF) que hace referencia a la identidad que ejecuta transacciones (*transactions*) en una red de negocio. Esta identidad está compuesta de un certificado digital y de una clave privada. 148

Integrated Development Environment (IDE) Entorno de desarrollo integrado; aplicación informática que proporciona servicios para facilitar al programador el desarrollo de *software*. XLIV, 217

Intensidad de Corriente Eléctrica Caudal de flujo de carga eléctrica (normalmente electrones) por unidad de tiempo que recorre un material. La unidad de resistencia en el Sistema Internacional (SI) es el amperio (A). 251

Inter-Integrated Circuit (I2C) Circuito Interintegrado; protocolo síncrono de comunicaciones en serie que emplea dos líneas para transmitir la información: una para los datos (*Serial Data* -SDA-) y otra para la señal de reloj (*Serial Clock* -SCL-). También es necesaria una tercera línea, pero esta sólo es la referencia (masa). Es muy usado en la industria, principalmente para comunicar microcontroladores y sus periféricos en sistemas integrados. XLIV, 28, 250, 259, 285

Internet of Things (IoT) Internet de las Cosas; concepto referido a la interconexión digital de objetos del mundo real con Internet, convirtiéndose así en objetos inteligentes de forma que adquieren la capacidad de transferir datos a través de la red, sin requerir de interacción humano a humano o humano a computador. V, VII, XLIV, 2, 16, 100, 131, 203, 275

iOS Sistema operativo creado por la compañía Apple Inc. para sus dispositivos móviles: Iphone, Ipad y Ipod. 218

ISO/IEC 27005:2008 Norma estándar de la Organización Internacional de Normalización (*International Organization for Standardization* -ISO-) y la Comisión Electrotécnica Internacional (*International Electrotechnical Commission* -IEC-) que proporciona directrices para el proceso de gestión de riesgos de seguridad de la información y sus actividades. 237

Java Lenguaje de programación de propósito general, concurrente, orientado a objetos, que fue diseñado por la empresa Sun Microsystem en 1991 específicamente para tener tan pocas dependencias de implementación como fuera posible con la intención de permitir que una misma aplicación pudiese ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura del dispositivo. 21, 139, 144, 217

Java ARchive (JAR) Archivo Java; archivo que permite ejecutar aplicaciones escritas en el lenguaje Java. XLIV

JavaScript (JS) Lenguaje de programación ligero, interpretado y dialecto del estándar ECMAScript que fue inventado por Brendan Eich y se utiliza principalmente en el lado del cliente (*front-end*) de sitios web. Entre otras características, se encuentran: orientación a objetos e imperativa, débilmente tipado y dinámico y basado en prototipos. XLIV, 24, 54

JavaScript Doc (JSDoc) Lenguaje de marcado utilizado para documentar el código fuente de ficheros JavaScript (JS). XLIV, 145

JavaScript Object Notation (JSON) Formato de texto ligero de intercambio de datos, legible para los seres humanos y fácil de interpretar y generar para las máquinas. Este formato está constituido por dos estructuras universales: una colección de pares de nombre/valor y una lista ordenada de valores. XLIV, 24, 109, 145

Kanban Metodología formulada por David J. Anderson para el desarrollo de proyectos que se centra en mejorar la visibilidad del flujo de trabajo utilizando el tablero Kanban el cual facilita la limitación del trabajo en curso con el fin de no sobrecargar a los miembros del equipo y cumplir con los plazos de entrega. 247

Lenguaje de Marcado También denominado como lenguaje de marcas, es un lenguaje para el procesamiento, definición y presentación de texto con el fin de que los sistemas informáticos puedan manipularlo de forma sencilla. Junto al texto, se incorporan también etiquetas que contienen información adicional acerca de la estructura del documento. El lenguaje de marcas más extendido es el HTML (*HyperText Markup Language*). 145

Ley de Ohm Ley básica de los circuitos eléctricos, postulada por el físico y matemático alemán Georg Simon Ohm, que establece que la diferencia de potencial (V) que aplicamos entre los extremos de un conductor determinado es proporcional a la intensidad de la corriente (I) que

circula por el citado conductor. Ohm completó la ley introduciendo la noción de resistencia eléctrica (R), originando la fórmula general de la ley de Ohm: $V = R \cdot I$. 251

Light-Emitting Diode (LED) Diodo emisor de luz; fuente de luz constituida por un material semiconductor dotado de dos terminales. XLIV, 37, 132, 218, 250, 254, 286

Linter Término que designa todas aquellas herramientas que realizan tareas de comprobación y análisis estático del código fuente con el fin de proporcionar información que posibilite la mejora de la calidad de éste. 145

Liquid Crystal Display (LCD) Pantalla de cristal líquido; pantalla delgada y plana formada por un número de píxeles en color o monocromos colocados delante de una fuente de luz o reflectora. XLIV, 37, 132, 218, 250, 254, 260, 263, 285

Macintosh Operating System (MacOS) Sistema operativo creado por la compañía Apple para sus equipos de sobremesa y portátiles. Está basado en Unix y usa HFS+ (*Hierarchical File System +*) para integrar un sistema de archivos propio. XLIV, 160, 219, 277

Markdown Lenguaje de marcado ligero creado por John Gruber que facilita la aplicación de formato y estilo a un texto empleando de forma especial una serie de caracteres. 143, 146

Minimum Viable Product (MVP) Producto que posee las suficientes características para satisfacer a los clientes iniciales y proporcionar retroalimentación para el desarrollo futuro y por tanto validar por lo menos una parte del negocio. XLIV, 158

Node Package Manager (NPM) Gestor de paquetes para JavaScript (JS), utilizado también por defecto para el entorno de ejecución Node.js. XLIV, 24, 145

Node.js Entorno de ejecución *open source*, multiplataforma, asíncrono y basado en eventos del lado del servidor para JavaScript (JS) que fue construido utilizando el motor V8, desarrollado por Google para uso de su navegador Google Chrome. 24, 145, 218

Open Source De código abierto; programa cuyo código fuente se encuentra disponible para el uso y/o modificación libre por parte de otros usuarios. Este tipo de *software* suele ser desarrollado por la colaboración pública y desinteresada de desarrolladores. V, VII, 3, 18–20, 23, 145

Paradigma Cliente-Servidor Patrón arquitectónico para el desarrollo de sistemas distribuidos basados en el concepto de diálogo petición-respuesta. Distribuye la aplicación en 2 tipos de entidades cada una de ellas presenta un rol bien diferenciado: cliente o servidor. 104

Participant Participante; concepto relativo a Hyperledger Composer (HC) y la Blockchain (BC) de Hyperledger Fabric (HF) que hace referencia a los miembros de una red de negocio los

cuales poseen activos (*assets*) y envían transacciones (*transactions*). Este concepto es común a otros tipos de Blockchain (BC). 140, 148

Patrón de Diseño Esqueleto de soluciones a problemas comunes en el desarrollo de *software*. Los patrones de diseño se clasifican en: patrones creacionales, patrones arquitectónicos o estructurales y patrones de comportamiento. 100, 102, 139

Patrón Modelo-Vista-Controlador (MVC) Estilo de arquitectura de *software* maduro e inventado en el contexto de Smalltak -lenguaje reflexivo de programación orientado a objetos- que, utilizando 3 componentes bien diferenciados, separa en una aplicación los datos de la lógica de la aplicación y de la lógica de la vista. XLIV, 100, 139

Peer-to-Peer (P2P) Red entre pares; tipo de arquitectura para la comunicación entre aplicaciones que permite a individuos comunicarse y compartir información con otros individuos sin necesidad de una autoridad central que facilite la comunicación. Este tipo de redes optimizan el uso de recursos. XLIV, 1, 8

Platform as a Service (PaaS) Plataforma como servicio; categoría de servicios en la nube que proporcionan una plataforma y un entorno que incluye todos los recursos *software* necesarios para soportar el ciclo de vida completo del desarrollo y puesta en marcha de aplicaciones (diseño, desarrollo, compilación, pruebas, distribución y administración, hospedaje, etc.), donde los usuarios pueden acceder a ellas simplemente a través de un navegador web. XLIV, 105, 137

Plugin Complemento *software* que contiene un grupo de funciones o características específicas para agregarselas a una aplicación. 60, 143, 218

Practical Extraction and Report Language (Perl) Lenguaje de programación de propósito general e interpretado diseñado por Larry Wall en el año 1987 con el objetivo principal de simplificar las tareas de administración de un sistema Unix. Hereda características y estructuras de otros lenguajes, como por ejemplo del lenguae C, del lenguaje interpretado Bourne Shell (sh), etc. 138

Proof of Concept (PoC) Prueba de concepto; implementación, a menudo resumida o incompleta, de un método o idea de producto en detalle realizada con el propósito de verificar que el concepto es susceptible de ser explotado de una manera útil. El objetivo principal es valorar el concepto de producto antes de comenzar su desarrollo a nivel técnico o físico lo que constituirá el producto final. V, VII, XLIV, 3, 14, 84, 131, 203, 275

Prototipo Representación inicial y limitada de un producto o sistema que posee las características de la versión final o parte de ellas y permite a las partes interesadas modelar el comportamiento en situaciones reales y explorar su uso, creando así un proceso de diseño

de iteración que genera calidad. 4, 70, 72, 83–85, 132, 137, 171, 172, 204, 217, 249–252, 254, 257, 262

Python Lenguaje de programación creado por Guido van Rossum. Se trata de un lenguaje *open source* de *scripting* independiente de la plataforma e interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible y simple de implementar. 28, 54, 70, 102, 138, 146, 147, 158, 163, 164, 169, 204, 218, 259, 260, 262, 264, 274, 276, 280, 285, 301

Query Concepto relativo a Hyperledger Composer (HC) y la Blockchain (BC) de Hyperledger Fabric (HF) que hace referencia al método para retornar datos de la Blockchain (BC), en concreto, la información almacenada en el *World State* o estado de la base de datos (BBDD). Este concepto es común para hacer referencia a la consulta realizada contra una BBDD. 146, 148

Random Access Memory (RAM) Memoria de acceso aleatorio; memoria principal de un dispositivo donde se almacenan datos y programas. Esta memoria es de tipo volátil lo que significa que los datos no se guardan de manera permanente, es por ello, que cuando deja de existir una fuente de energía en el dispositivo la información se pierde. XLV, 219

Raspberry Pi (RPi) Computador de placa reducida (*Single Board Computer* -SBC-) de bajo coste desarrollado en el Reino Unido por la Fundación Raspberry PI en 2011, con la misión de estimular y fomentar la enseñanza de las ciencias de la computación en las escuelas, aunque no empezó su comercialización hasta el año 2012. V, VII, XLV, 4, 28, 105, 132, 158, 218, 225, 250, 252, 257, 274, 275

Representational State Transfer (REST) Transferencia de estado representacional; conjunto de técnicas orientadas a crear servicios web en los que se renuncia a la posibilidad de especificar la interfaz de los servicios de forma abstracta a cambio de contar con una convención que permite manejar la información mediante una serie de operaciones estándar. XLV, 137, 295

Resistencia Componente electrónico diseñado para introducir una resistencia eléctrica (oposición al flujo de electrones al moverse a través de un conductor) determinada entre dos puntos de un circuito eléctrico. 250, 251

Resistencia Eléctrica Oposición al flujo de electrones al moverse a través de un material conductor. La unidad de resistencia en el Sistema Internacional (SI) es el ohmio (Ω). 251

Rivest, Shamir y Adleman (RSA) Sistema criptográfico asimétrico o de clave pública desarrollado en 1977. Es el primer algoritmo de este tipo y el más utilizado siendo válido tanto para cifrar como para firmar digitalmente. La seguridad de este algoritmo radica en el problema de la factorización de números enteros grandes con sus números primos. XLV, 265

Scaffolding Técnica para la generación automática de código a partir de una plantilla preestablecida. 139

Script Programa, usualmente simple y de tamaño pequeño, que permite realizar tareas específicas a partir de un conjunto de órdenes definidas. Este tipo de programa no es compilado, sino que es interpretado línea a línea en tiempo real durante su ejecución para codificar la información y traducirla a lenguaje máquina. Su principal utilidad es interactuar con el sistema operativo de manera automatizada, aunque muchas veces se utilizan lenguajes interpretados, como Perl o Python, para realizar tareas más complejas. 138, 275

Secure Hash Algorithm (SHA) Algoritmo de Hash Seguro; familia de funciones *hash* de cifrado publicadas por el Instituto Nacional de Normas y Tecnología (*National Institute of Standards and Technology* -NIST-) de EE.UU. Existen varias versiones, desde la primera versión SHA-0 creada en 1993 hasta la más reciente SHA-3 publicada en 2012. Esta última versión se caracteriza por ser la que más difiere de sus predecesoras siendo el descubrimiento de vulnerabilidades la razón de la existencia de varias versiones. XLV, 260

Secure Shell (SSH) Protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura cliente-servidor y que permite a los usuarios conectarse de forma encriptada a un host remoto. XLV, 258

Secure Sockets Layer (SSL) Capa de puertos seguros; protocolo criptográfico que proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía. Habitualmente, solo el servidor es autenticado (es decir, se garantiza su identidad) mientras que el cliente se mantiene sin autenticar. XLV, 153, 265

Sensor Dispositivo diseñado para detectar información de una magnitud externa y y transformarla en otra magnitud, normalmente eléctrica, que sea posible cuantificar y manipular. 4, 36–38, 61–63, 103, 132–134, 140, 141, 218, 234, 250, 254, 255, 260, 275, 285–287, 290, 294, 298

Shell Intérprete de comandos; Programa que provee una interfaz de usuario para acceder a los servicios del sistema operativo facilitando la forma en que se invocan los distintos programas disponibles en el dispositivo informático. 138

Single Board Computer (SBC) Computador de placa única; placa de tamaño reducido que contiene todos o la mayor parte de los componentes de un ordenador: microprocesador, memoria RAM (*Random Access Memory*), dispositivos de entrada/salida (E/S), etc. XLV, 132, 161, 218

Sistema de Control de Versiones Método para controlar y registrar las diferentes versiones por las que pasará un archivo o conjunto de archivos a lo largo del tiempo, de modo que permite la recuperación de versiones específicas en un futuro. 217

Sistema Operativo (SO) Conjunto de programas de un sistema informático encargados de controlar y gestionar los procesos básicos y recursos hardware. Además, permite el funcionamiento de otros programas. XLV, 160, 218

Software as a Service (SaaS) *Software* como servicio; modelo de distribución de *software* en el que tanto el *software* como los datos manejados son centralizados y alojados en un servidor externo al usuario y gestionados por el proveedor del servicio por lo que es éste el encargado de garantizar la disponibilidad, seguridad, mantenimiento, soporte, etc. de la herramienta. XLV, 137

Solid-State Drive (SSD) Unidad de estado sólido; tipo de dispositivo de almacenamiento de datos que utiliza memoria no volátil para almacenar datos, en lugar de los discos magnéticos de las unidades de discos duros convencionales HDD (*Hard Disk Drive*). XLV, 219

Spring *Framework* de código abierto y de propósito general creado por Rod Johnson para el desarrollo de aplicaciones en el entorno Java, principalmente. 143

Stakeholder En gestión de proyectos, todas aquellas personas u organizaciones que afectan o son afectadas por el proyecto, ya sea de forma positiva o negativa. 100, 157, 222

System Management Bus (SMBus) Bus de Administración del Sistema; subconjunto del protocolo I2C (*Inter-Integrated Circuit*). XLV, 259

Tensión eléctrica También conocido como voltaje, es una magnitud física que cuantifica la diferencia de potencial eléctrica entre dos puntos. La unidad de resistencia en el Sistema Internacional (SI) es el voltio (V). 250

Token Cadena de caracteres que tiene un significado coherente en cierto lenguaje de programación. Suelen ser utilizados como identificadores, para el proceso de autenticación, etc. 33, 35, 52, 60, 66, 78, 109, 137, 144, 147, 152, 173, 189, 190, 294

Transaction Transacción; concepto relativo a Hyperledger Composer (HC) y la Blockchain (BC) de Hyperledger Fabric (HF) que hace referencia al procedimiento por el cual los participantes (*participants*) interactúan con los activos (*assets*). Este concepto es común a otros tipos de Blockchain (BC). VII, 103, 140, 148

Transport Layer Security (TLS) Seguridad de la capa de transporte; protocolo criptográfico definido por primera vez en 1999 que garantiza comunicaciones seguras por una red, comúnmente Internet. Se basa en las especificaciones previas de SSL (*Secure Sockets Layer*) por lo que se considera un protocolo más estable y más seguro que éste. XLV, 155, 265

Unified Modeling Language (UML) Lenguaje unificado de modelado; lenguaje de modelado para documentar la arquitectura, el diseño y la implementación de sistemas *software*, tanto en estructura como en comportamiento. XLV, 100, 145

Uniform Resource Locator (URL) Localizador de recursos uniforme; secuencia de caracteres que sigue un estándar y que permite denominar recursos dentro del entorno de Internet para que puedan ser localizados. XLV, 35, 143, 295

Unix Sistema operativo portable, multitarea y multiusuario, desarrollado en el año 1969 por un grupo de empleados (Ken Thompson y Dennis Ritchie) de los laboratorios Bell de la compañía AT&T. Fue creado como un sistema operativo propietario para manejar servidores. 138

User Interface (UI) Medio con que el usuario puede comunicarse con una máquina, computadora o dispositivo, y comprende todos los puntos de contacto entre el usuario y el sistema. XLV, 51, 100, 158, 338

Virtual Machine (VM) Máquina virtual; *software* que permite emular el funcionamiento de un sistema de computación y ejecutar programas como si fuese un sistema real con la salvedad de que los componentes son virtuales. XLV, 105, 137, 217, 258, 267

Voltage at the Common Collector (VCC) Voltaje de corriente continua; pin de entrada de potencia principal de un dispositivo que suele ser mayor a 5 voltios (V) en circuitos lógicos típicos. El voltaje es más alto con respecto a la conexión a tierra (*Ground* -GND-). XLV, 250

Web Application Archive (WAR) Archivo de aplicación web; archivo JAR (*Java ARchive*) utilizado para distribuir una colección de clases Java, *servlets*, librerías de etiquetas, páginas web estáticas, etc. que juntos constituyen una aplicación web. XLVI, 258

Wireless Fidelity (WiFi) Fidelidad inalámbrica; tecnología que permite la interconexión inalámbrica de dispositivos electrónicos basados en los estándares IEEE 802.11. Inicialmente fue creado para acceder a redes locales inalámbricas, aunque actualmente es utilizado frecuentemente para establecer conexiones a Internet. XLVI, 250

YAML Ain't Markup Language (YAML) Formato estándar, propuesto por Clark Evans en 2001, de serialización de datos legible por los seres humanos para todos los lenguajes de programación. XLVI, 143, 260

Acrónimos

AES Advanced Encryption Standard. *Glosario:* AES. 266,

API Application Programming Interface. *Glosario:* API. 24, 137, 147, 149, 241, 265, 295,

ASCII American Standard Code for Information Interchange. *Glosario:* ASCII.

AUP Agile Unified Process. 27, 221–224,

Bash Bourne-again Shell. *Glosario:* Bash. 54, 147, 218, 259, 275,

BBDD Base de Datos. 2, 10, 21, 30, 32, 33, 54, 73, 109, 110, 132, 133, 137, 140, 143, 144, 172, 218, 260, 265, 291, 293, 297, 298, 328, 332,

BC Blockchain. *Glosario:* BC. V, VII, 1–21, 23–26, 31, 35, 55, 58, 60, 62, 63, 73, 84–86, 105, 109, 110, 131, 133–135, 137–141, 144, 146, 147, 152, 176, 199, 203, 204, 225, 261, 262, 264, 275, 295, 296, 298, 302, 309, 310, 318–320, 328–330, 332, 333,

CA Certificate Authority. *Glosario:* CA. 153, 264, 268–272,

CSI Camera Serial Interface. *Glosario:* CSI. 250,

CSR Certificate Signing Request. *Glosario:* CSR. 267,

CSS Cascading Style Sheets. *Glosario:* CSS. 54, 143,

DBaaS Database as a Service. *Glosario:* DBaaS. 109, 137,

DLT Distributed Ledger Technology. *Glosario:* DLT. V, VII, 8, 18, 23,

GND Ground. *Glosario:* GND. 250,

GPG GNU Privacy Guard. *Glosario:* GPG. 31, 32, 39, 42, 74, 133, 135, 146, 147, 178, 183, 204, 259, 260, 264, 291, 298, 301, 302, 306, 308,

GPIO General Purpose Input Output. *Glosario:* GPIO. 218, 250–252, 260, 286, 288,

GSP Groovy Server Pages. *Glosario:* GSP. 144,

HC Hyperledger Composer. *Glosario:* HC. 3, 21, 23–25, 110, 137–139, 144, 147, 152, 153, 176, 199, 204, 217, 218, 261–263, 265–267, 295, 296, 310, 328,

HF Hyperledger Fabric. *Glosario:* HF. V, VII, 3, 4, 19, 20, 22–25, 55, 105, 131, 134, 135, 137, 139, 144, 146, 148, 176, 199, 217, 225, 261, 262, 275, 295, 298,

HTML HyperText Markup Language. *Glosario:* HTML. 54, 146,

HTTP Hypertext Transfer Protocol. *Glosario:* HTTP. 105, 148, 152, 191, 260,

HTTPS Hypertext Transfer Protocol Secure. *Glosario:* HTTPS. 56, 105, 148, 152, 191, 258, 260, 263, 265, 267, 269–272, 297,

I2C Inter-Integrated Circuit. *Glosario:* I2C. 28, 37, 38, 84, 250, 259, 261, 263, 285–287, 290,

IDE Integrated Development Environment. *Glosario:* IDE. 217, 218,

IoT Internet of Things. *Glosario:* IoT. V, VII, 2–4, 16–18, 73, 83, 84, 100, 109, 131, 132, 146, 203, 204, 262, 275, 285, 297, 310, 327,

JAR Java ARchive. *Glosario:* JAR.

JS JavaScript. *Glosario:* JS. 24, 54, 143, 145,

JSDoc JavaScript Doc. *Glosario:* JSDoc. 145,

JSON JavaScript Object Notation. *Glosario:* JSON. 24, 109, 145, 149,

LCD Liquid Crystal Display. *Glosario:* LCD. 37, 38, 43, 55, 72, 73, 84, 132, 147, 218, 234, 250, 254, 260, 263, 285–287, 290,

LED Light-Emitting Diode. *Glosario:* LED. XIV, 37, 43, 55, 132, 218, 250, 251, 254, 256, 286,

MacOS Macintosh Operating System. *Glosario:* MacOS. 160, 219, 269, 277,

MVC Modelo-Vista-Controlador. *Glosario:* MVC. 100, 101, 103, 139,

MVP Minimum Viable Product. *Glosario:* MVP. 158,

NPM Node Package Manager. *Glosario:* NPM. 24, 145,

P2P Peer-to-peer. *Glosario:* P2P. 1, 8, 16–18,

PaaS Platform as a Service. *Glosario:* PaaS. 105, 137,

PoC Proof of Concept. *Glosario:* PoC. V, VII, 3, 4, 14, 19, 84, 131, 203, 204, 275,

RAM Random Access Memory. *Glosario:* RAM. 219,

REST Representational State Transfer. *Glosario:* REST. 24, 137, 147, 295,

RPi Raspberry Pi. *Glosario:* RPi. V, VII, XIII, 4, 28, 29, 53, 55, 57, 68–70, 72, 83, 89, 105, 112, 123, 132–135, 137, 138, 146, 147, 158, 161, 162, 165, 167–169, 218, 225, 234, 250–252, 257, 259–264, 275–277, 280, 282, 285, 288, 298, 301, 302,

RSA Rivest, Shamir y Adleman. *Glosario:* RSA. 265,

RUP Rational Unified Process. 221,

SaaS Software as a Service. *Glosario:* SaaS. 137,

SBC Single Board Computer. *Glosario:* SBC. 132, 161, 218,

sh Bourne Shell. *Glosario:* sh. 138,

SHA Secure Hash Algorithm. *Glosario:* SHA. 8, 133, 153, 260, 267,

SMBus System Management Bus. *Glosario:* SMBus. 259,

SO Sistema Operativo. *Glosario:* SO. 160, 218, 219, 269,

SSD Solid-State Drive. *Glosario:* SSD. 219,

SSH Secure Shell. *Glosario:* SSH. 258,

SSL Secure Sockets Layer. *Glosario:* SSL. 153, 155, 265, 268,

TFM Trabajo Fin de Máster. 139, 217, 218, 222, 224, 226, 232,

TI Tecnologías de la Información. 223, 224,

TLS Transport Layer Security. *Glosario:* TLS. 155, 265, 270,

TPC Tercera Parte de Confianza. V, VII, 1, 5, 6, 131, 133,

UI User Interface. *Glosario:* UI. 51, 100, 101, 103, 158, 338,

UML Unified Modeling Language. *Glosario:* UML. 100, 106, 145,

URL Uniform Resource Locator. *Glosario:* URL. 35, 143, 147, 150, 295, 310,

VCC Voltage at the Common Collector. *Glosario:* VCC. 250,

VM Virtual Machine. *Glosario:* VM. 105, 137, 139, 217, 258, 267,

WAR Web Application Archive. *Glosario:* WAR. 258, 265,

WiFi Wireless Fidelity. *Glosario:* WiFi.

YAML YAML Ain't Markup Language. *Glosario:* YAML. 143, 145, 146, 260, 263,

Capítulo 1

Introducción y objetivos

Todo proyecto requiere de una descripción previa que sitúe al lector en el panorama general del mismo, con el fin de que éste llegue a comprender con perfección las ideas y metas que se persiguen con su desarrollo. En este capítulo se aborda de forma general los principales rasgos de este documento.

1.1. Definición del problema

Internet ha cambiado la forma de vida y la sociedad en general. Nuestra actividad diaria depende cada vez más de la información que se obtiene de Internet, de forma que es necesario contar con mecanismos que permitan dirimir si los datos que se obtienen son fiables. Esta cuestión se ha resuelto tradicionalmente a través de alguna suerte de Tercera Parte Confiable (TPC). Sin embargo, la participación de intermediarios también presenta desventajas, entre las que se encuentra la posible degradación de la propia privacidad si la TPC accede, sin el consentimiento expreso, a la información sensible y personal.

Una posible solución es la descentralización de la gestión de la información, de forma que no es necesario una autoridad central intermediaria que tenga acceso a los datos. Aquí, es donde nace el concepto de cadena de bloques o Blockchain (BC) [38], un libro de registros o transacciones (*ledger*) distribuido y descentralizado, del que todos los que participan en la red almacenan una copia que se actualiza mediante un protocolo *peer-to-peer* (P2P) de consenso y que habilita una red a priori sin confianza a ser completamente funcional y con la misma seguridad que una totalmente confiable. En este contexto, el protocolo de consenso distribuido se constituye en garante de la integridad de la información y, por ende, de su veracidad.

En su origen, BC se inventó para sustentar Bitcoin [63], la primera criptomoneda -medio digital de intercambio- descentralizada no emitida por un banco central [22]. Sin embargo, su aplicación no queda limitada a las criptomonedas. En efecto, la BC de Bitcoin habilita la escritura de in-

formación no vinculada a transacciones (por ejemplo, mediante el campo *OP_RETURN*). Esta información puede ser utilizada como canal de control de trazabilidad de las cadenas de producción (p.ej. Everledger [19]), de certificación de documentación (como es el caso de Stampery [79] y MaidSafe [58]), de hipotecas (p.ej. Zensar [89]), de títulos o cualquier otro documento oficial (p.ej. Bitfury [7], Factom [21], ChromaWay [11] y Velox.re [84] son ejemplos de *start-ups* que ofrecen este tipo de servicio), así como aplicaciones de control de integridad de información en ámbitos relacionados con la seguridad lógica, destacando el proyecto Guardtime [27] el cual cuenta con el patrocinio de la Agencia de Proyectos de Investigación Avanzados de Defensa, DARPA [14].

Uno de los ámbitos donde BC tiene especial interés es en el Internet de las Cosas (*Internet of Things* -IoT-) [10], [15]. La IoT constituye una red de dispositivos físicos que por naturaleza se conectan entre sí e intercambian datos para hacer nuestras vidas más sencillas y eficientes. Sin embargo, cada dispositivo puede ser cualquier *cosa* desde una televisión, un vehículo, un aspirador o hasta un frigorífico y todos funcionan de forma diferente y con diferentes niveles de seguridad implementados. Esta variabilidad en las interfaces de acceso a la información y los mecanismos de intercambios de datos, introduce una incertidumbre en lo referente a las diversas fuentes de datos. Aquí es donde entra en juego la BC, en específico aquellas evoluciones de la BC de Bitcoin mediante la incorporación de los denominados *smart contracts* o *chaincodes*[1] -concepto introducido por Nick Szabo en 1996 [80]- y de modelos de control de acceso.

Motivado por la reciente explosión de interés de esta tecnología y este ámbito, en este proyecto se examina si la combinación de BC-IoT proporciona ventajas frente a otras combinaciones a la hora de la securización de un entorno IoT ya que no siempre la transición o uso de una red descentralizada tiene sentido y aunque tuviese sentido podría darse el caso de que los requerimientos de la aplicación no se adaptasen a una red de este tipo. La securización del entorno se efectúa mediante la trazabilidad de los sucesos que se produzcan en él de forma que si ocurre un suceso no previsto pueda ser detectado el motivo y origen de éste. Esta prueba o evidencia, consistente de una grabación del entorno en el momento en el que se produjo la incidencia y que permite detectar falsos positivos, debe ser salvaguardada de cualquier alteración no autorizada y para ello se utiliza el método de almacenamiento de BC. Otra solución podría ser utilizar un método tradicional de almacenamiento como, una base de datos (BBDD) o un servicio de almacenamiento en la nube aunque en estas situaciones no se podría garantizar la integridad del contenido original de la evidencia.

[1]Se trata de los *smart contract* de Ethereum. Consultar [18] para obtener más información.

1.2. Estudio de implementaciones similares

El contexto en el que se ubica el presente proyecto, aunque es de un amplio interés, es tan novedoso y concreto que actualmente no existe o no se ha encontrado una implementación final o una prueba de concepto (*Proof of Concept* -PoC-) con las mismas o similares líneas de trabajo -securización de un entorno con BC- y tecnologías.

Sí es verdad que, a raíz de este enorme interés por parte del ecosistema tecnológico y empresarial, han surgido y están surgiendo numerosas soluciones ubicadas en muy diferentes sectores, algunas de ellas propietarias mientras que otras *open source*, que aplican BC al entorno IoT. Entre ellas, se pueden encontrar soluciones para gestionar el acceso a dispositivos IoT [66], para preservar la privacidad en transacciones de sistemas energéticos [54], para comunicar los *Smart Things* -pequeñas cosas inteligentes situadas en dispositivos IoT- [77], para trazar productos de una cadena de suministro (calidad del mismo, prueba de origen, etc.) [2], para proveer analíticas inteligentes y datos del entorno en tiempo real con el objetivo de conocer, por ejemplo si se está realizando una construcción ilegal o existen partículas inusuales en el aire [65], para digitalizar el mundo físico y poder llevar un control de personas y activos, de experiencia del cliente, etc. [52].

1.3. Objetivo general

A resultas de la actual carencia de la existencia de implementaciones que aborden la problemática de cómo asegurar la veracidad de una manera irrefutable de sucesos no controlados en entornos IoT se propone el proyecto actual empleando las herramientas citadas en el anexo A y con el que se consigue satisfacer los siguientes objetivos:

- Comprender el concepto de BC (origen, conceptos clave, modo de funcionamiento, taxonomía, beneficios frente al almacenamiento de información tradicional, etc.) y todo el ecosistema que engloba.
- Revisión del estado actual de la tecnología BC y qué ha supuesto en los diferentes sectores.
- Analizar cuáles son las principales implementaciones de BC hoy en día, sus características, ventajas y limitaciones.
- Conocer qué es Hyperledger [29] y comprender el motivo de su aparición.
- Conocer los proyectos existentes en Hyperledger y analizar con mayor profundidad la tecnología de Hyperledger Fabric (HF) [35] y Hyperledger Composer (HC) [31] asimilando la jerga de las que está compuesta.
- Aprender a desplegar una BC permisionada en HF y desarrollar *chaincodes*.

- Entender qué es el IoT y cómo puede mejorar la trazabilidad de los sistemas y la seguridad al combinarse con la BC.
- Aprender conceptos relacionados con el área de la electrónica para poder construir un prototipo *hardware*.
- Desplegar una configuración IoT mediante una Raspberry Pi (RPi) y diversos sensores y dispositivos de entrada y salida (E/S).
- Desarrollar una PoC sobre HF que permita garantizar la veracidad e integridad de sucesos anómalos registrados en entornos IoT.

1.4. Estructura del documento

El presente documento presenta una estructura dividida en siete capítulos:

- **Capítulo 1 - Introducción y objetivos:** presenta la introducción al actual proyecto, exponiendo la motivación y objetivos así como el material y herramientas empleadas para su implementación.
- **Capítulo 2 - Estado del arte:** proporciona al lector un acercamiento al contexto y marco tecnológico que engloba a este proyecto junto a la justificación de su utilización sobre otras tecnologías que podrían producir resultados similares.
- **Capítulo 3 - Análisis:** reúne los requerimientos, funcionalidades y características que el sistema debe implementar para satisfacer las necesidades del cliente.
- **Capítulo 4 - Diseño:** modela el diseño y la arquitectura que debe poseer el sistema gracias a la información obtenida durante la etapa análisis.
- **Capítulo 5 - Implementación:** revela los entresijos de cómo el proyecto ha sido implementado.
- **Capítulo 6 - Casos de prueba:** describe los casos de prueba ejecutados para garantizar la calidad final de la solución.
- **Conclusiones y trabajo futuro:** concluye resumidamente el trabajo realizado y hace alusión a las diferentes líneas que puede tomar el proyecto como trabajo futuro.
- Para terminar, existen una serie de anexos con información de interés que abordan desde la metodología de desarrollo *software* implementada junto con aspectos relacionados con la planificación (organización interna, plan de fases e iteraciones, estimación de costes y gestión de riesgos) hasta la descripción del prototipo construido y los manuales de instalación y configuración y de usuario.

Capítulo 2

Estado del arte

Conocer los inicios, pilares y fundamentos básicos sobre los que se sustenta la tecnología principal del proyecto es esencial para entender el contexto y la jerga que engloba. En este capítulo se aborda un breve estudio sobre la tecnología de cadena de bloques o Blockchain (BC) para que el lector pueda adquirir un conocimiento general y superficial.

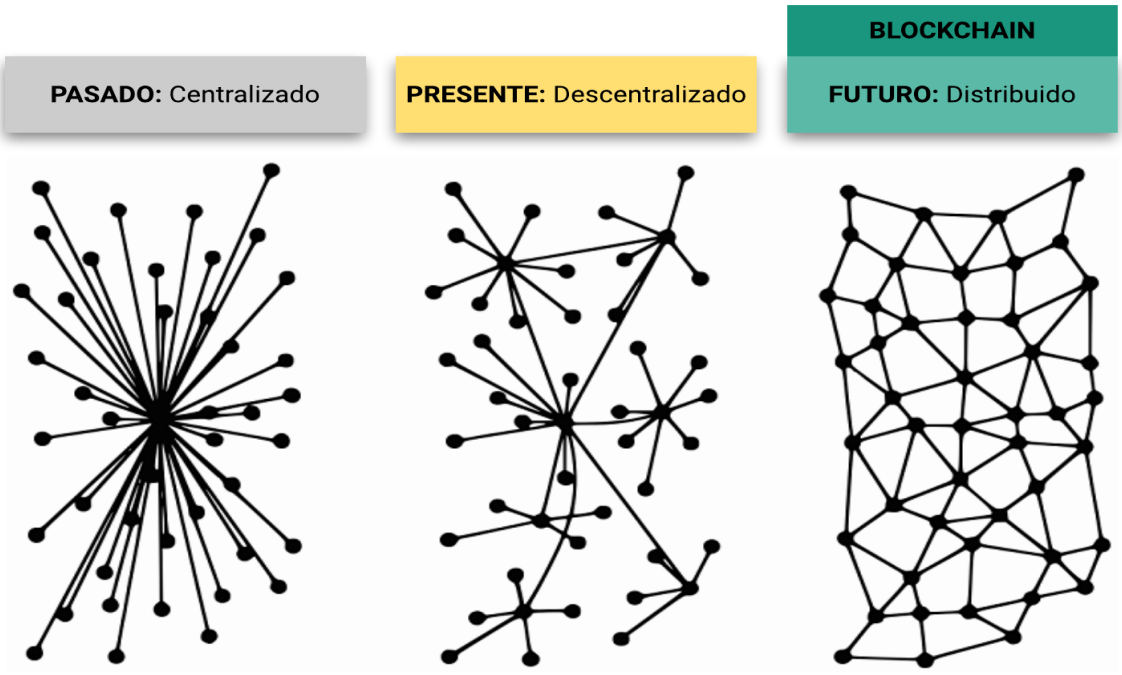
2.1. Introducción

Los servicios tecnológicos de hoy en día y los sistemas en general en los que se basa el funcionamiento de la sociedad y el mundo económico y financiero siguen la filosofía tradicional articulada bajo la confianza en terceras partes. Un ejemplo muy a la mano del día a día es la funcionalidad ofrecida por entidades bancarias respecto a la gestión de las cuentas de clientes. El banco, que actúa como una tercera parte de confianza (TPC), tiene la responsabilidad de garantizar la veracidad de la información personal asociada a dicha cuenta, de las transferencias emitidas y recibidas y del saldo disponible en ella. Los clientes son por tanto la parte que emite y entrega la confianza a las entidades bancarias por lo que quedan a merced del buen hacer de éstas. Como garante de la integridad y veracidad de los datos, estas entidades han desarrollado grandes sistemas con un enfoque normalmente centralizado por lo que únicamente una autoridad central es la encargada de gobernar y velar por la coherencia y seguridad de la información existiendo de tal manera puntos únicos de fallo.

Estos sistemas, considerados como complejos y costosos, no han tenido en cuenta en general dentro de sus objetivos principales facilitar la interoperabilidad y colaboración con otros sistemas. La gran cantidad de información que almacenan y manejan provoca que sea inviable y nada fácil de conseguir una transferencia de información eficiente y rápida entre distintos sistemas e incluso organizaciones por lo que estos datos se pueden encontrar en múltiples silos de información, impactando negativamente en el rendimiento y en la efectividad a la hora de trabajar, tomar decisiones o incluso gestionar áreas o unidades de negocio. Frente a este escenario, surgieron sistemas don-

de la información se localiza de manera descentralizada en múltiples puntos de almacenamiento omitiendo la existencia de un almacenamiento central y por tanto el punto único de fallo. Sin embargo, la falta de una interconexión total mediante una red de comunicaciones desemboca en que no sea adecuado para determinadas áreas donde se requiere interoperabilidad. Aquí, es donde las propiedades de BC entran en juego ya que habilitan nuevos escenarios totalmente distribuidos gracias a la transformación de la topología de las redes económicas y sociales, tal y como muestra la Figura 2.1.

Figura 2.1: Topologías de red.



BC es un nuevo paradigma llamado a liderar el tránsito de una Internet de la Información (en el que la información puede ser replicada indefinidamente) hacia un Internet del Valor. En esta nueva Internet los activos de información se comparten de una manera segura y descentralizada, son únicos y tienen una historia trazable que está almacenada en un *ledger* (libro de contabilidad o de registros) distribuido. Esta transición implica modificar los modelos existentes en muchos negocios con la definición de nuevas plataformas distribuidas en las que la figura de los intermediarios o TPC pierde vigencia en favor de planteamientos de consenso y confianza distribuida. En primer lugar, el modelo de confianza distribuido deparado por la tecnología de consenso de la BC supone un cambio fundamental respecto a la figura de la TPC. La existencia de una tecnología que permite la creación de redes totalmente distribuidas en las que sus participantes no necesitan una autoridad central o de confianza representa un giro radical en el diseño de nuevos modelos de sistemas ya que reduce la fricción en las operaciones y disminuye las barreras de colaboración. Por otra parte, el modelo distribuido en el que todos los participantes mantienen una copia del *ledger*,

hace que la información se encuentre repartida y replicada. Este factor, unido a la forma en al que se construye el consenso entre los nodos de transferencia y almacenamiento de información, garantiza la inmutabilidad e integridad de los registros y da lugar a sistemas resilientes altamente transparente y auditables y con gran tolerancia a fallos, a la vez que acaba con el problema de los silos de información. En definitiva, se habilita la instauración de una *fuente única de verdad*, que es común a todas las organizaciones involucradas, además de compartida y supervisada por todas ellas.

Sin embargo, este nuevo paradigma no siempre es válido o es el más adecuado para implementar en un proyecto ya que una solución tradicional puede ser perfectamente lo que se necesita. El caso de uso, los participantes y los objetivos perseguidos son los determinantes de qué red o plataforma es la más idónea para hacer realidad la solución de negocio necesaria y para ello existen cinco preguntas que se deberían formular antes de la toma de decisión [91]:

1. ¿Hay más de una organización, área o unidad de negocio formando parte de la red? Este primer criterio incide sobre la característica fundamental de BC con la que hace posible la colaboración y trabajo conjunto.

2. ¿Se requiere algún tipo de consenso para la funcionalidad que se necesita? El consenso forma parte intrínseca de una BC y establece las reglas de juego para validar transacciones y proporcionar así confianza entre los participantes.

3. ¿Es necesaria la trazabilidad de algún tipo de activo? Uno de los principales casos de uso de una BC es la trazabilidad de un activo digital o físico digitalizado.

4. ¿Se necesita que los datos manejados por la solución sean inmutables? Es decir, una vez registrados es obligatoriamente necesario evitar cualquier alteración no autorizada.

5. ¿Es importante disponer de una *fuente única de verdad*? En las soluciones tradicionales donde existe colaboración pueden existir múltiples almacenamientos que den lugar a todo tipo de conflictos como, por ejemplo la divergencia de datos sobre un activo o transacción.

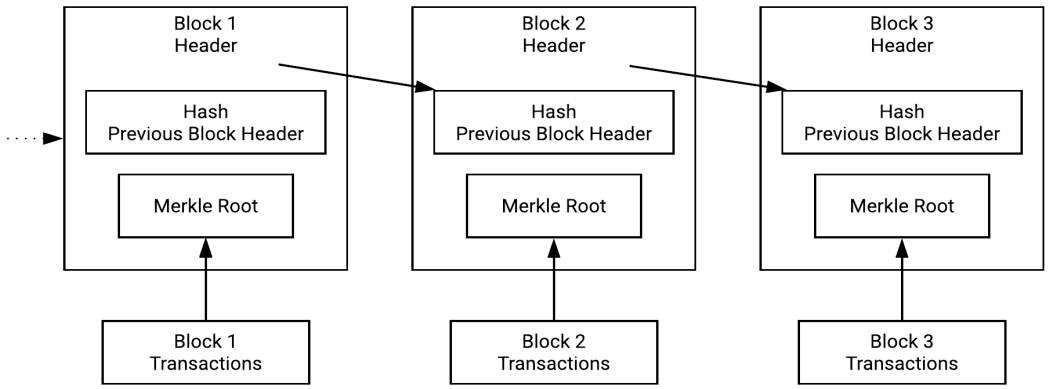
Para que una idea de negocio sea considerada como un buen caso de uso para la tecnología BC debería cumplir obligatoriamente la pregunta 1 y al menos responder afirmativamente a una de las preguntas restantes.

2.2. Concepto de Blockchain

La cadena de bloques o Blockchain (BC) es un concepto amplio cuya definición no se encuentra del todo cerrada. No obstante, se pueden establecer una serie de elementos comúnes a las diversas

variantes existentes. Se puede considerar como un tipo de tecnología de registro distribuido (*Distributed Ledger Technology* -DLT-) [85], es una estructura de datos distribuida, también conocida como libro de registros o *ledger*, que es replicada[1] y compartida entre todos aquellos miembros (nodos) que pertenezcan a una red P2P (*peer-to-peer*), es decir, una red distribuida. Este *ledger* contiene todas las transacciones registradas en la red estableciendo a quién pertenece qué. Las transacciones en sí no son almacenadas de forma aleatoria e individual, si no que son agrupadas en bloques y es el propio bloque el que finalmente se registra en la cadena junto con información adicional, vinculando uno con otro (Figura 2.2) para facilitar la recuperación de la información y asegurar la no alteración de la información ya que una posible modificación en el contenido de una transacción requiere la perturbación de cada bloque de la cadena para ocultar esta manipulación. Cuando una transacción es registrada en la BC se considerada válida y tras ciertas inclusiones posteriores de bloques es considerada como inmutable.

Figura 2.2: Estructura de una cadena de bloques.



Cada bloque en la cadena se identifica por un valor *hash* criptográfico generado por medio de aplicar dos veces una función hash a la cabecera del bloque con el algoritmo SHA-256. Además, contiene la siguiente información (Figura 2.3):

- Cabecera de bloque conteniendo metadatos sobre el mismo como, por ejemplo la versión, la marca de tiempo (*timestamp*), el valor *hash* del bloque anterior, un valor *nonce*[2] para efectos de minería y el valor *hash* de la raíz del árbol de Merkle.
- Árbol de Merkel el cual es una estructura de datos que sintetiza las transacciones contenidas en el bloque. Esta estructura es construida recursivamente aplicando funciones hash a pares de transacciones hasta obtener un único valor *hash*, llamado raíz o raíz de Merkle (Figura 2.4).

[1]No necesariamente tiene que ser replicada por todos los nodos. En algunas BC un puede ser ligero, de forma que localmente solo posee una versión reducida de la cadena.

[2]Número arbitrario de un solo uso.

Figura 2.3: Contenido de un bloque.

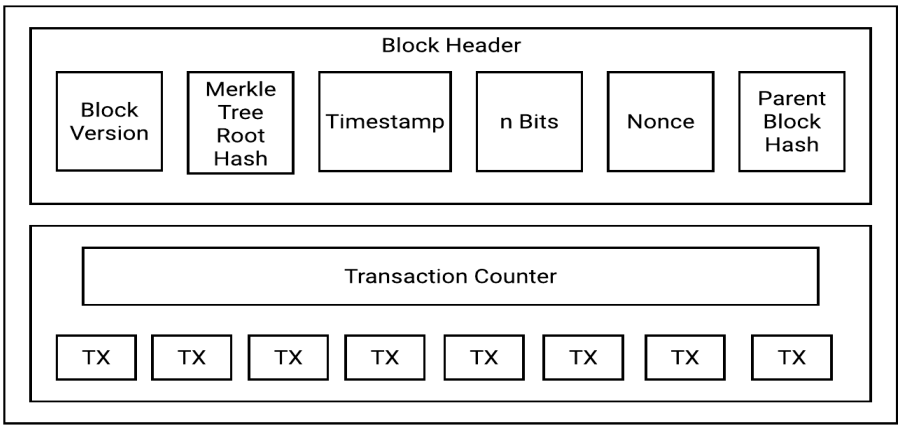
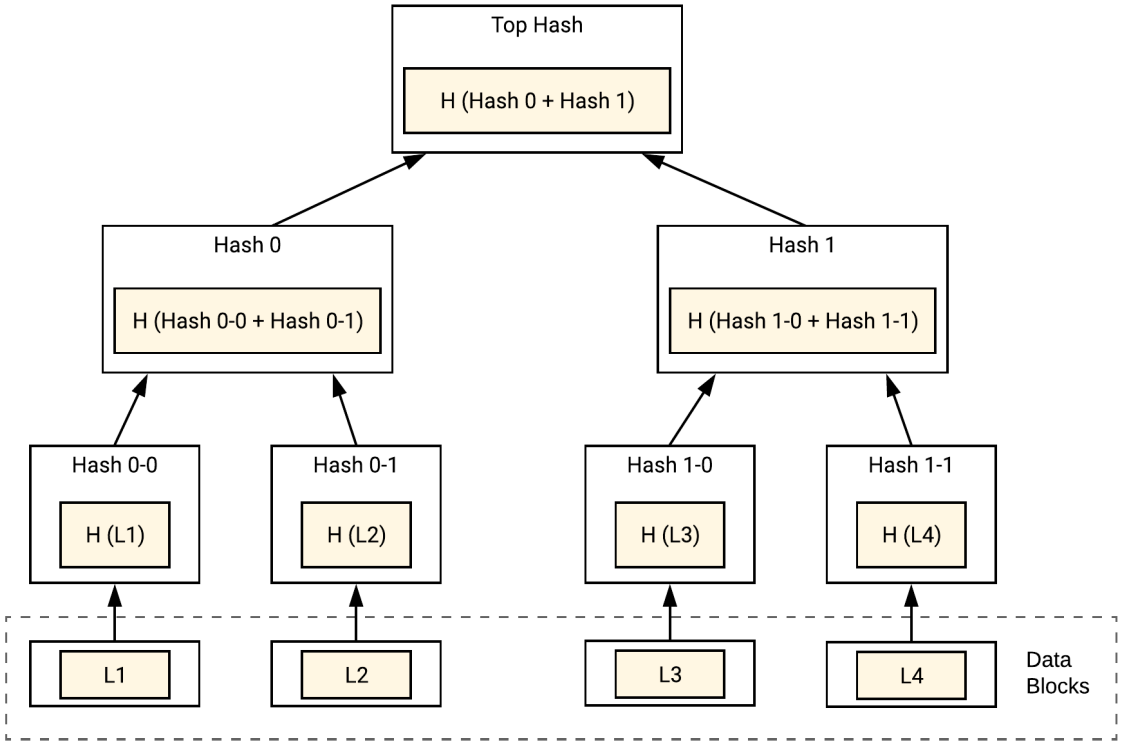


Figura 2.4: Estructura del árbol de Merkle.



Cualquier nodo o *peer* con acceso a la BC puede leer los bloques y el conjunto de transacciones que alberga y descubrir cuál es el estado mundial de los datos que se intercambian en la red. Cada nodo, que puede considerarse como punto de entrada para varios usuarios diferentes de BC en la red, contiene una copia ordenada y actualizada del *ledger*, donde:

1. Cada usuario o nodo individual interactúa con la BC a través de su par de claves, firmando

las transacciones propias con la clave privada y direccionándolas en la red a través de la clave pública[3]. El uso de criptografía asimétrica proporciona autenticación, integridad y no repudio a la red.

2. Los nodos vecinos se encargan de validar cada transacción entrante antes de retransmitirla para que aquellas que sean consideradas inválidas puedan ser descartadas y no afecten negativamente al estado de la red. Cuando la transacción es validada se extiende a toda la red.

3. Todas las transacciones que han sido recopiladas y validadas por la red utilizando el proceso anterior durante un intervalo de tiempo acordado, se ordenan y empaquetan en un bloque candidato junto con la marca de tiempo. Este es un proceso llamado minería. El primer nodo que consigue formar un bloque candidato lo transmite a la red donde la elección de este nodo y el contenido del bloque depende del mecanismo de consenso que se emplee.

4. El resto de nodos verifican que el bloque sugerido a ser almacenado en la BC contenga transacciones válidas y referencia vía *hash* al bloque previo correcto de la cadena. Si ese es el caso, se agrega el bloque a la cadena y aplica las transacciones que contiene para actualizar el estado del mundo. En caso contrario, el bloque propuesto se descarta, marcando como el final de una ronda.

La etapa de validación de transacciones es esencial para otorgar a la red un estado consistente y sin alteraciones ya que ésta puede estar gestionado por parte no confiables. Para prevenir un posible caos proveniente de este entorno distribuido y para ayudar a la red a alcanzar un estado global adecuado, cada red BC necesita establecer ciertas reglas para determinar si una transacción entrante es considerada válida o no. La validez suele venir de la mano de la firma presentada en la transacción la cual debe coincidir con el usuario origen de dicha transacción.

2.2.1. Principales características

La principales características que proporciona esta tecnología son:

- *Ledger* o base de datos (BBDD) distribuida y protegida criptográficamente en la que cada nodo tiene una copia de toda la información almacenada.
- Red de confianza. Esta confianza reside en el consenso y acuerdo alcanzado entre todos los participantes, sin que exista la figura de un tercero que controle y gobierne todo lo que sucede.
- La actividad de la red (transacciones) es transparente, verificable y auditable por cualquier usuario en BC públicas y por usuarios identificados en BC privadas.

[3]Dependiendo de la implementación, la clave pública puede ser la dirección o un código *hash* de esta dirección.

- Inmutabilidad. Una vez introducida la información, ésta no se puede modificar por parte de ningún actor implicado ni ningún administrador.
- Paradigma de colaboración con lo que se reduce la fricción entre los distintos participantes en base a unas reglas bien definidas y de fácil adopción, sin necesidad de un gran nivel de acuerdo.

2.2.2. Criptografía

La tecnología de BC está construida sobre dos conceptos claves de criptografía: función hash y firma digital. Las funciones hash son funciones matemáticas con las siguientes propiedades [59]:

- La entrada de la función es una cadena de cualquier tamaño.
- La salida de la función tiene siempre un tamaño fijo por lo que ante un mínimo cambio en la entrada, la salida varía significativamente.
- Es eficiente de computar. La complejidad computacional en el peor caso es $\mathcal{O}$(n).
- Es resistente frente a colisiones lo que implica que no es factible encontrar un número x e y, tal que H(x) = H(y) siendo H la función hash.

Por su parte, la firma digital es un método criptográfico que permite garantizar la autoría e integridad de la información por medio de asociar la identidad de un usuario o sistema a un mensaje o documento.

2.2.3. Consenso

Las firmas digitales son utilizadas para verificar que una transacción es firmada por quién dice ser el firmante. Sin embargo, aquí surge el problema de que cualquiera puede enviar una misma transacción o criptomoneda -moneda digital- dos veces, cada una con una firma totalmente válida. En los sistemas centralizados, la autoridad central es la gobernante de la red y por tanto la controladora de prevenir el problema del doble gasto [50]. En un sistema totalmente descentralizado, los participantes de la red, independientemente de que tenga plena confianza o no entre ellos, necesitan acordar las reglas de ciertos principios y funcionalidades que serán comunes para todos y de validez de transacciones para prevenir este problema. El protocolo de consenso distribuido es empleado para conseguir este fin, donde se tiene en cuenta el número de nodos que forman la red y un número arbitrario de ellos que podrían ser maliciosos. Este consenso por tanto asegura que:

- Todos los nodos honestos acuerden un único valor.
- Este valor debe ser originado por un nodo honesto.

En la actualizad existen multitud de algoritmos de consenso donde el uso de uno u otro depende de las necesidades requeridas. Los más conocidos son:

- Algoritmo de prueba de trabajo (*Proof of Work* -PoW-): protocolo donde se utiliza el poder computacional del *hardware* para resolver cálculos matemáticos y determinar quién es el creador de un nuevo bloque el cual recibe una recompensa en forma de una nueva criptomoneda. Tiene como principal objetivo detener los ciberataques como los ataques de denegación de servicios (DDoS)[4]. Este concepto no es nuevo [64] y la idea subyacente fue publicada por Cynthia Dwork y Moni Naor en 1993 [17].
- Algoritmo de prueba de participación (*Proof of Stake* -PoS-): protocolo cuya idea fue sugerida en el foro de *bitcointalk* en el año 2011 y que se basa en la idea de seleccionar de forma determinista el creador de un nuevo bloque en base a su riqueza o participación. En este protocolo la recompensa recibida es la comisión pagada por cada transacción.

2.2.4. Smart Contracts

El concepto de *smart contract* o contrato inteligente [55] hace referencia a un acuerdo entre dos o más partes donde se define lo que se puede hacer, cómo se puede hacer, qué pasa si algo no se hace, etc. que es capaz de ejecutarse y hacerse cumplir por sí mismo, de manera autónoma, automática y sin intermediarios. Estos contratos aplicados a la BC son programas informáticos escritos con algún lenguaje de programación que ejecutan acuerdos establecidos entre dos o más partes haciendo que ciertas acciones sucedan como resultado de que se cumplan una serie de condiciones específicas, implicando un avance al permitir mayor funcionalidad distribuida que la hasta entonces utilizada en BC, pagos descentralizado.

Este concepto no es reciente si no que llevan desarrollándose desde el año 1996, cuando el famoso criptógrafo Nick Szabo acuñó el término por primera vez. Nick propuso este sistema de contratos por aquel entonces, sin embargo la infraestructura tecnológica del momento lo hacía inviable. El *smart contract* se puede decir que vive en una atmósfera no controlada por ninguna de las partes implicadas en el contrato en un sistema descentralizado aunque esté almacenado en cada uno de ellos. Esto significa que:

1. Se programan las condiciones.
2. Se firman por ambas partes implicadas.
3. Se despliega en una BC para que no pueda ser modificado.
4. Se ejecuta desembocando en un resultado inmutable y totalmente transparente.

[4]Ataque cuya finalidad es saturar los recursos de un sistema al enviar múltiples peticiones falsas.

Como beneficios principales se pueden encontrar los siguientes:

1. Implementar un estado de seguridad mayor al del contrato tradicional debido a que estos contratos se almacenan de forma encriptada y permite la interacción entre personas que no se conocen entre sí sin que haya riesgo de estafa alguno.

2. Reducir costes al no ser necesario una persona o entidad que valide el contrato.

3. Reducir el tiempo asociado a este tipo de interacciones al automatizar las tareas.

4. Ampliar el horizonte de los modelos de negocio.

2.2.5. Taxonomía

El gran auge de la tecnología BC ha supuesto un crecimiento elevado de redes basadas en ésta así como su diversidad de características y funcionalidades que implementan. El diseño de una taxonomía permite categorizar las redes BC y su forma de uso con el fin de facilitar la tarea de evaluación, comparación y toma de decisiones a la hora de implementar sistemas basados en esta tecnología. A continuación, se expone una posible taxonomía [87], [90]:

- Quién tiene acceso a la red. Cualquier usuario puede unirse y acceder a la red (BC pública o no permisionada, p.ej. Bitcoin [63], Ethereum [18] o Litecoin [56]) o por el contrario existe un control de acceso, ya sea a través de autenticación, listas de control de acceso (ACL), etc., para poder unirse a la red (BC privada o permisionada, p.ej. Hyperledger [29], R3 [72] o Ripple [75]). Además, ha proliferado otro tipo de BC, conocida como híbrida (p.ej. BigchainDB [6] y Evernym [20]) la cual es una combinación de las dos anteriores ya que existe un control de acceso pero una vez accedido todas las transacciones son públicas.

- Quién puede registrar transacciones o minar. Principalmente aplica a BCs de carácter privado puesto que todo participante conectado posee un identidad única y por tanto pueden existir roles que diferencien quién puede desplegar *smart contracts*, quién posee el privilegio de registrar transacciones [32] o quién puede participar en el proceso de minado [62].

- Arquitectura de descentralización. Esta característica está directamente relacionada con la categorización en función del privilegio de acceso a la red puesto que las BC públicas son completamente descentralizadas (el consenso es alcanzado por todos los nodos sin que haya una confianza plena entre ellos), mientras que las BC híbridas y privadas son parcialmente centralizadas o centralizadas, respectivamente (el consenso es alcanzado únicamente por un grupo de nodos seleccionados confiando por tanto el resto de nodos en ese grupo seleccionado).

- Quién puede tomar parte del proceso de consenso. Todos los nodos son capaces de participar en el proceso de consenso y por tanto validar transacciones (BC públicas) o solamente un subconjunto seleccionados de nodos puede participar (BC híbridas y privadas). Debido al ataque Sybil [16], donde los atacantes crean gran cantidad de nodos anónimos con el fin de corromper el sistema e influir en las decisiones tomadas de forma distribuida, los consensos en redes públicas son computacionalmente costosos incentivando de forma económica -criptomonedas- a los mineros que mantienen la red mientras que las redes privadas al operar en un entorno regulado y controlado con nodos identificados pueden emplear consensos más livianos al estar exentos de este tipo de ataque.

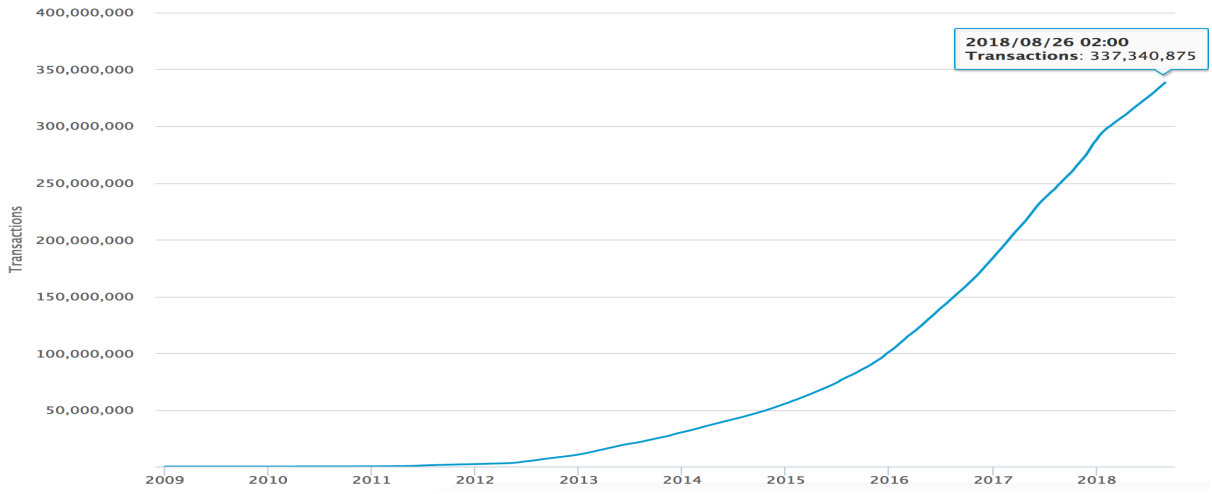
- Tipología de las transacciones [88], basadas en el modelo UTXO o *Unspent Transaction Output* (transacciones de estilo Bitcoin) o basadas en el modelo cuenta-balance (*smart contracts* de Ethereum). En el primer modelo, cada transacción únicamente puede gastar una cantidad límite procedente de una transacción no gastada anterior generando así una nueva transacción no gastada con una cantidad resultante de tal manera que la cartera (*wallet*[5]) de un usuario contiene una lista de transacciones no utilizadas asociadas con todas las direcciones propiedad del usuario que resultarán en el balance global de la cuenta. Este modelo favorece la escalabilidad al poder ejecutar transacciones en paralelo y la privacidad debido al uso de nuevas direcciones para cada transacción. Mientras que el modelo cuenta-balance mantiene el balance de cada cuenta como un estado global comprobando en cada transacción que dicho balance sea mayor a la cantidad que se desea transferir. Por su parte, este modelo es más simple y eficiente.

2.3. Revisión del estado actual de la tecnología Blockchain

Tecnológicamente hablando, BC es incipiente pero está creciendo a gran velocidad y ya existen numerosas soluciones tecnológicas que derivan total o parcialmente su lógica de negocio a esta capa de persistencia ya sea en BCs de carácter público o privado, éstas últimas orientadas a soportar aplicaciones en el ámbito corporativo y poniendo el foco sobre la privacidad de la información y el rendimiento. La creación de consorcios tanto nacionales como internacionales, así como la proliferación de pruebas de concepto (*Proof of Concept* -PoC-), pilotos, plataformas y servicios en gran variedad de sectores muestra cómo la capacidad transformadora de BC ya está impactando en los negocios y el foco ya está puesto en cómo materializar los beneficios de este paradigma. Pero esta tecnología no solamente se emplea para la trazabilidad de activos en diferentes sectores, su primera implementación fue para originar un sistema de pago de gestión no centralizado (Bitcoin) a través de una criptomoneda. Aunque actualmente sea principalmente de carácter especulativo, la Figura 2.5 muestra este crecimiento experimentado en el número de transacciones totales realizadas así como el número de usuarios que poseen un *wallet* BC (Figura 2.6).

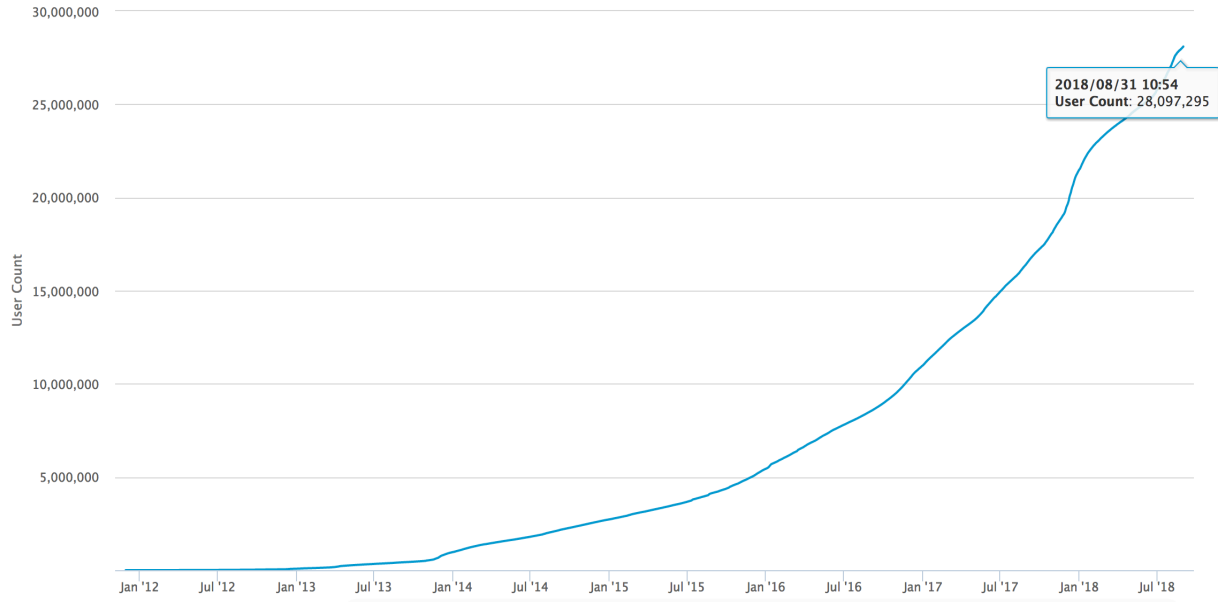
[5]Monedero digital para gestionar criptomonedas.

Figura 2.5: Número total de transacciones Bitcoin.



Fuente de comprobación: `https://www.blockchain.com/es/charts/`

Figura 2.6: Número total de wallets Blockchain.



Fuente de comprobación: `https://www.blockchain.com/es/charts/`

BC posee el potencial necesario para transformar todos los negocios y actualmente su aplicación se está explorando en la práctica totalidad de sectores. Aunque en un primer momento el sector financiero fue el abanderado de la transformación apostando por ella y explorando sus capacidades con los pagos transacionales gracias al ahorro de costes y agilidad de los procesos,

actualmente en todos los ámbitos hay un gran interés. El ámbito de la logística y el transporte no es ajeno al desarrollo de iniciativas en torno a BC y, aunque se encuentra en una fase inicial respecto a otros sectores, se empieza a observar una actividad relevante y la expectativa es que ésta se incremente a corto plazo. A día de hoy, esta tecnología ofrece un sinfín de oportunidades y se están llevando a cabo ya varios proyectos referentes al comercio internacional, al transporte terrestre de mercancías, a la trazabilidad de productos y cadena de suministro, a la automoción con modelos de aseguradoras *pay-as-you-drive*, a la gestión de identidad, al Internet de las Cosas (*Internet of Things* -IoT-), etc.

El potencial impacto de BC exige a todos los actores del sector un posicionamiento al respecto, así como una valoración rigurosa de las amenazas y oportunidades que puede introducir para evitar quedarse al margen en la transformación del negocio y poder así dar respuesta a los nuevos modelos que puedan surgir y definir un planteamiento estratégico que les permita obtener ventajas competitivas. Este análisis debe partir de la comprensión de las implicaciones que BC representa en cuanto a nuevos modelos de desintermediación y los beneficios que aporta en estos nuevos escenarios en relación a la confianza, la transparencia, la trazabilidad y la automatización de operaciones.

2.4. Principales implementaciones de Blockchain

El paradigma BC puede considerarse que surge de la mano de Bitcoin en un enfoque de red pública en el año 2009 por Satoshi Nakamoto, red en la cual no hay restricciones para sumar nuevos nodos y cualquiera puede utilizar un cliente para unirse y empezar a participar en la red. Este protocolo y red P2P consensuada que se utiliza como sistema de pago a través de su criptomoneda bitcoin -primera criptomoneda por capitalización en el mundo- se caracteriza por ser descentralizado y no estar respaldado ni controlado por ningún gobierno ni banco central, fin para el que surgió junto a las necesidades de imposibilitar la falsificación o duplicación, a la irreversibilidad en las transacciones (inmutabilidad) y a la perseveración de la identidad (pseudoanónimo).

Para alcanzar el consenso entre todos los nodos que integran la red no confiable intercambiando información y que puede estar potencialmente comprometida se utiliza el algoritmo PoW, basado en la idea del algoritmo Hashcash [28] para evitar el correo *spam*. Este algoritmo de consenso se usa para confirmar transacciones y producir nuevos bloques en la cadena donde los mineros, encargados de velar por la seguridad y la no alteración de la red, compiten entre ellos para completar transacciones en la red mediante cálculos matemáticos (función *hash*, factorización de enteros, protocolo de rompecabezas guiado, etc.), fáciles de comprobar y cuya dificultad es ajustada cada cierto tiempo en base a la cantidad de usuarios, la potencia actual y la carga de la red y obtener así recompensas en forma de bitcoins a la vez que se introducen nuevas monedas en la red, utilizando

para ello el poder computacional del *hardware*. Con su uso se resuelve el problema de los ataques ataques de denegación de servicio (DDoS) y el problema los generales bizantinos [68], juego de lógica donde en un asedio hay un número indeterminado de generales que debe coordinarse para la conquista y únicamente uno, el comandante, cursa la orden (atacar o retirarse) siendo el resto tenientes. En este grupo de generales puede existir uno o más traidores con el objetivo de que no se cumpla la orden.

El gran problema que presenta Bitcoin, a parte del gran gasto energético que requiere hoy en día debido al algoritmo de consenso utilizado [49], es la limitación a ser utilizado como medio de pago o almacén de valor. Ethereum desarrollada por la fundación Ethereum con base en Suiza fue concebido originalmente como una versión mejorada de la criptomoneda Bitcoin, para superar las limitaciones de su lenguaje de programación, proporcionando características avanzadas tales como la creación y ejecucción de aplicaciones en la BC y, lo que es más importante, establecer acuerdos o contratos inteligentes. Esta tecnología posee una BC pública, permitiendo también la creación de privadas que utilizan el mismo protocolo y la misma base tecnológica. La red pública posee una criptomoneda nativa, el Ether, que es actualmente la segunda criptomoneda por capitalización en el mundo. Pero realmente no destaca por la moneda virtual en sí, si no por el potencial para aumentar sustancialmente la eficiencia de la industria computacional, y proporcionar un gran impulso a otros protocolos P2P adicionando por primera vez una capa económica.

2.5. Blockchain orientado al entorno IoT

El IoT se refiere a la interconexión de numerosos objetos físicos del mundo real los cuales son provistos de una conexión a Internet. Tales dispositivos, cuyo número se encuentra en plena expansión -se predice una cantidad 20 billones para el año 2020-, adquieren información sobre el entorno y se comunican con otros dispositivos de este tipo o sistemas *software* a través de Internet. Como consecuencia de esta interacción enriquecida producen gran cantidad de datos que pueden ser consumidos por diferentes servicios lo que proporciona numerosos beneficios. A pesar de ello, también abre la puerta a numerosos problemas de seguridad y privacidad en parte debido a la transferencia de información que puede ser considerada como personal y sensible y que puede revelar comportamientos y preferencias de los propios usuarios.

Este paradigma que está creando nuevos desafíos tecnológicos y ofreciendo ventajas competitivas en múltiples sectores está variando la forma en la que la comunicación tiene lugar. Donde antes era requerido una interacción explícita para la toma de datos, ahora esos datos pueden ser solicitados y obtenidos sin intervención humana alguna y en numerosos casos en un dominio no controlado ni confiable. La protección de estos datos a lo largo de su ciclo de vida es evidente, ya que la *International Data Corporation* (IDC) estima que el 90 % de las organizaciones que

implantan este ecosistema son más vulnerables a sufrir ataques a través de estos dispositivos.

La privacidad se encuentra en riesgo cuando la información es gestionada por una compañía o autoridad central la cual puede hacer un uso ilegítimo de ésta. Con el propósito de prevenir esta situación, la implantación de la tecnología BC con una red P2P fomenta un diseño privado, descentralizado y sin puntos únicos de fallos donde la privacidad y seguridad es garantizada por lo que los datos recolectados del entorno son garantes de su veracidad e inmutabilidad. Esta combinación BC-IoT [51], [12] permite la creación de una infraestructura distribuida segura sobre la que transaccionar abriendo nuevos horizontes para plantear sistemas en los que los distintos dispositivos se comuniquen más directamente. Estas operaciones, que gracias a los contratos inteligentes pueden ser más complejas, son registradas y autenticadas independientemente de su finalidad (creación, modificación o eliminación) en el *ledger* asegurando así cualquier intento de alteración no autorizada y habilitando la confianza sobre un entorno a priori no confiable. Este nuevo enfoque proporciona las siguientes ventajas frente a un entorno centralizado:

- Mejora la seguridad al evitar información incosistente gracias al consenso de todos los nodos de la red que tienen una réplica de la BC.
- Mejora la gestión de la identidad de los dispositivos.
- Aporta transparencia y automatización de los procesos al poder ser trazables.
- Reduce costes al fomentar la interacción digital y la colaboración.
- Elimina posibles cuellos de botella cuando numerosas peticiones son realizadas y existe un control de acceso.

2.6. Hyperledger

Lo cierto es que las BC públicas no son tan descentralizadas y transparentes como proclaman sus creadores y defensores [5]. Además, los *smart contracts* de Ethereum tienen múltiples problemas de seguridad [4]. Por eso, y por la escalabilidad y otros temas legales y normativos, tiene sentido considerar otros modelos de BC tal como los contenidos en Hyperledger.

Hyperledger es un consorcio de carácter colaborativo y de código abierto (*open source*) anunciado en diciembre del año 2015 por la fundación Linux -organización referente a nivel tecnológico sin ánimo de lucro creada a principios de 2010 para promover y acoger proyectos *open source*- para investigar y evolucionar la tecnología DLT, en concreto la BC de uso privado y orientada al ámbito económico y empresarial donde gracias a la capacidad de ejecución de transacciones privadas y a la inclusión de novedosas características se reconducirá la forma en la cual los negocios y procesos comerciales tienen lugar. El objetivo de esta iniciativa, que trabaja de la mano con la

Enterprise Ethereum Alliance (EEA), es establecer los estándares para garantizar la interoperabilidad y estabilidad de los diferentes proyectos y así evitar los riesgos asociados a un ecosistema fragmentado, junto con la idea de promover gran variedad de negocios basados en esta tecnología con un amplio abanico de soluciones y usos diferentes.

Esta línea de trabajo es posible al esfuerzo, conocimiento y a la dedicación por parte de numerosos socios y miembros, cuyo número cada vez es mayor, donde se incluyen compañías centradas en la BC (Blockchain, R3, etc.), compañías tecnológicas (IBM, Intel, Red Hat, VMware, Cisco, etc.), compañías financieras (J.P. Morgan, SWIFT, etc.) y entidades bancarias (BBVA), entre otras. Treinta miembros iniciales fueron los originarios de comenzar con esta andadura con dos de los proyectos actualmente en desarrollo donde en numerosas ocasiones son los propios desarrolladores y compañías las que desean ceder el proyecto para su incubación con el propósito de ganar visibilidad y favorecer su impulso al generar una comunidad a su alrededor. Este es el caso de Monax Industries (Hyperledger Burrow) y Sovrin (Hyperledger Indy), dos de los proyectos actualmente en desarrollo. El primer proyecto en entrar en fase de incubación -fase de prueba, desarrollo y difusión- fue el introducido por la compañía tecnológica IBM, el cual recibió posteriormente el nombre de Hyperledger Fabric (HF), y actualmente se encuentra en un estado activo, siendo también el primer proyecto en adquirir esta condición.

Aunque la estandarización es uno de los principales objetivos de Hyperledger, también se pueden encontrar otros no menos importantes como:

- Proporcionar una infraestructura neutral, abierta e impulsada por la comunidad y respaldada por una gobernanza técnica.
- Difundir la transferencia de conocimiento sobre esta tecnología y sus oportunidades en el contexto empresarial existente.
- Crear *frameworks open source* que permitan soportar operaciones de negocio multisectoriales.
- Desarrollar comunidades técnicas que definan casos de uso y desarrollen PoC.

Estos objetivos son alcanzados gracias al enfoque modular que proporciona Hyperledger donde se ubican todos los componentes que lo conforman actualmente, donde:

- En primer nivel se encuentra la infraestructura que hace referencia al ecosistema que difunde el aceleramiento del desarrollo abierto y la adopción comercial junto con la cobertura técnica, legal, *marketing* y organizacional que se proporciona.
- En segundo nivel se localizan los *frameworks* que proporcionan los elementos necesarios (*ledger*, algoritmo de consenso, contratos inteligentes, etc.) para el despliegue de la BC.

- En último nivel se ubican todas aquellas herramientas auxiliares que complementan a los *frameworks* ya sea en el despliegue, mantenimiento o diseño y prototipado de las redes BC.

2.6.1. Hyperledger Fabric

HF es el proyecto más conocido dentro de la iniciativa Hyperledger. Es *open source* y está orientado al desarrollo de soluciones empresariales que implementen la tecnología BC [30], [3]. Originado en un primer momento por la contribución de las compañías Digital Asset Holdings e IBM y actualmente en constante evolución[6] con diversas empresas involucradas y con una hoja de ruta interesante [36], ofrece diversas características (arquitectura modular y extensible, red transaccional de alto rendimiento, privacidad e identidad, escalabilidad, proveedores de membresía y algoritmos de consenso configurables, etc.) que tienden a mejorar muchos aspectos de productividad y fiabilidad y que lo distinguen de otras alternativas de BC.

Pero donde realmente este *framework* se diferencia, a parte de la ausencia de una criptomoneda, es en su BC de carácter privado y permisionado ya que permite controlar la sensibilidad sobre qué y cuánta información los participantes pueden compartir en una red de negocio a negocio (*Business-to-Business* -B2B-), es decir, permite delimitar quién posee acceso a la red y a qué activos y con qué privilegios, obligando de tal forma a que los participantes posean identidades[7] conocidas a través de la autenticación frente a un determinado proveedor de servicios de membresía confiable (*Membership Service Provider* -MSP-)[8], componente de confianza que especifica las reglas de membresía para una organización con el fin de gestionar y verificar la veracidad de una identidad y por tanto de autenticar en la red a los participantes.

La privacidad y confidencialidad de las transacciones se obtiene a través de los canales. En una red BC de HF pueden coexistir múltiples canales privados simultáneamente donde cada uno contiene su propio *ledger* con toda la información para funcionar correctamente, incluyendo transacciones, información del canal, miembros, etc. Esta información es invisible e inaccesible para cualquier participante de la red que no posea permiso de acceso para ese determinado canal. Esta característica es de especial interés en el ámbito empresarial donde se quiere por ejemplo ocultar la información de transacciones para que no sean públicas a otros participantes ya sea porque son diferentes clientes, pertenecen a distintas áreas de negocio o incluso porque almacenan información sensible de dicho participantes, etc.

El *ledger* en HF, donde cada participante o nodo de la red almacena una copia actualizada

[6] HF v1.2.0 fue liberado en la fecha 03 julio 2018.

[7] Cada identidad es encapsulada en un certificado digital X.509.

[8] HF habilita la arquitectura modular donde se permite entre otros aspectos implementar diferentes algoritmos de identidad o usar diferentes MSPs.

de éste de cada canal al que pertenezcan, está formado por la combinación de los dos siguientes componentes:

- Estado del mundo: describe el estado del libro de registros en un determinado punto temporal. Este componente es considerado como la BBDD y por defecto es de tipo clave-valor, LevelDB.
- *Log* de transacciones: se considera el historial de actualizaciones del componente anterior por lo que almacena todas las transacciones que ocurren (valor anterior y posterior) dando lugar a un determinado valor resultante en el estado del mundo.

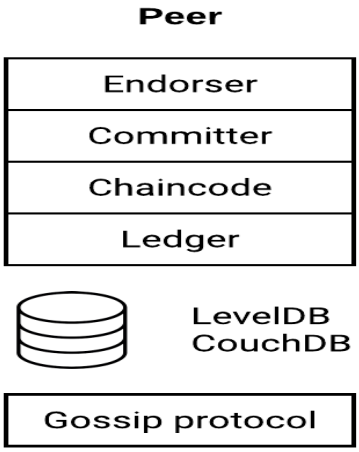
Esta información almacenada y considerada inmutable proviene de la ejecución de la lógica del sistema por parte de una solución empresarial externa, es decir, de la invocación de *chaincodes* o *smart contracts* cuando ésta necesita interactuar con el *ledger*, en ocasiones solamente con el componente estado del mundo. Estos contratos inteligentes que definen las reglas de las transacciones pueden ser implementados nativamente en diversos lenguajes donde actualmente Go y Java están soportados o implementados con Javascript utilizando la herramienta Hyperledger Composer (HC).

La red BC de esta tecnología está constituida por organizaciones que tienden a establecer el consenso, considerados miembros en el caso de formar parte de la construcción y mantenimiento de la red. Cada organización es responsable de configurar los *peers* que van a participar y su rol proveyendo en tal caso el material criptográfico necesario para ello. Pero no todos los nodos son del mismo tipo. Hay tres tipos de nodos con diferentes roles donde pueden contener los componentes de la Figura 2.7.

- *Endorser peer*: aquellos que reciben la invocación de una petición de transacción desde el cliente y aprueban o no la transacción en función de:
  - Verificar que la propuesta se encuentra bien formada.
  - Comprobar que no ha sido subida en el pasado (protección contra el ataque de replicación).
  - Validar la firma que contiene la transacción usando el MSP.
  - Verificar que el remitente está autorizado a realizar la transacción en el canal.
  - Ejecutar el *chaincode* simulando el resultado de la transacción pero sin actualizar el *ledger*.
- *Anchor peer*: aquellos que habilitan la comunicación entre *peers* de la misma organización o de diferentes, recibiendo y emitiendo las actualizaciones.

- *Orderer peer*: aquellos encargados de crear los bloques, ordenarlos cronológicamente por cada canal, entregarlos a todos los *peers* y persistir la información en el componente del estado del mundo actualizando así el *ledger* (*committer*).

Figura 2.7: Componentes de un peer.



El consenso en HF es efectuado por los *orderer peers* donde se pueden usar múltiples elecciones como, Solo (para redes en pruebas), PBFT (*Practical Byzantine Fault Tolerance*), SBFT (*Simplified Byzantine Fault Tolerance*, Kafka ordering service, etc. algunos aún en desarrollo. Éste último es el método empleado por defecto para redes en producción el cual se basa en la votación permisionada y ofrece tolerancias a fallas pero no tolerancia a fallas bizantinas el cual previene al sistema de alcanzar un acuerdo en caso de nodos maliciosos o defectuosos. Este proceso de consenso está formado por tres fases: validación de transacciones, fase de ordenación de transacciones válidas y validación de bloques y de la exactitud de los resultados.

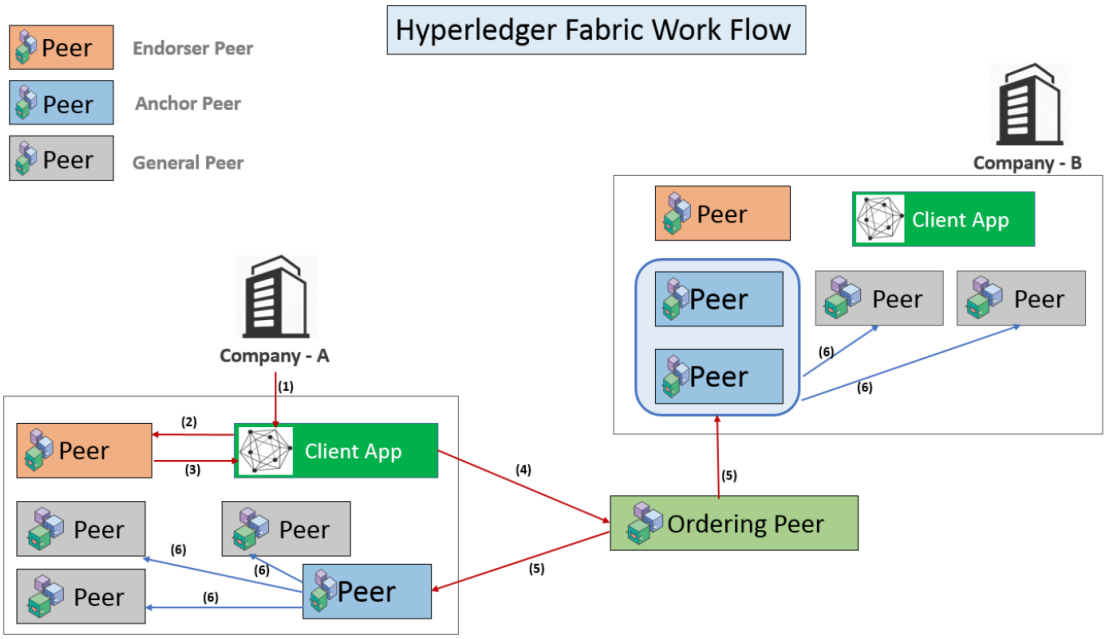
A modo resumen, la Figura 2.8 muestra el flujo de ejecución de una transacción cuando es ejecutada por un cliente, donde:

- Un participante miembro en la organización invoca una solicitud de transacción a través de una aplicación cliente.
- El cliente emite la solicitud de invocación de transacción a uno o varios *endorser peers*, en función de la configuración establecida en la organización.
- Cada *endorser peer* realiza varias comprobaciones y simula la ejecución del *chaincode*, regresando al cliente los resultados firmados por el propio *peer* y la aprobación o denegación de la transacción.

- Cliente envía la transacción aprobada al *orderer peer* para que sea ordenada e incluida en un bloque junto a su escritura en el *ledger*. Este bloque es emitido al resto *anchor peers* de las organizaciones miembro de la red.

- Los *anchor peers* emiten el bloque al resto de *peers* de la organización para que actualicen su *ledger* con este último bloque de forma que la red está sincronizada.

Figura 2.8: Flujo de ejecución de transacción en Hyperledger Fabric.



Fuente:
https://medium.com/coinmonks/how-does-hyperledger-fabric-works-cdb68e6066f5

2.6.2. Hyperledger Composer

HC es una herramienta *open source*, perteneciente también a la iniciativa Hyperledger, para desarrollar redes de negocio (*Business Network Definition* -BND-) de BC de soluciones construidas sobre la infraestructura de HF. Su finalidad no es otra que acelerar y facilitar el proceso de definición de la BND y del desarrollo de los *chaincodes* en una DLT. Si bien es cierto que esta red puede desarrollarse de forma nativa con la propia tecnología HF en lenguaje Go, también es cierto que implica el conocimiento de detalles de bajo nivel involucrados en las redes BC lo que ralentiza a su vez el desarrollo y puesta en producción de soluciones descentralizadas. Este conjunto de abstracciones que ofrece para modelar e integrar la solución BC provoca que el desarrollador tome

un menor tiempo y esfuerzo para la definición del modelo y que únicamente se centre en la lógica de la red.

Esta herramienta está construida con JavaScript (JS), aprovechando herramientas modernas como Node.js, NPM (*Node Package Manager*), etc. y el desarrollo del componente principal, la BND, implica la definición utilizando el lenguaje de modelado CTO de:

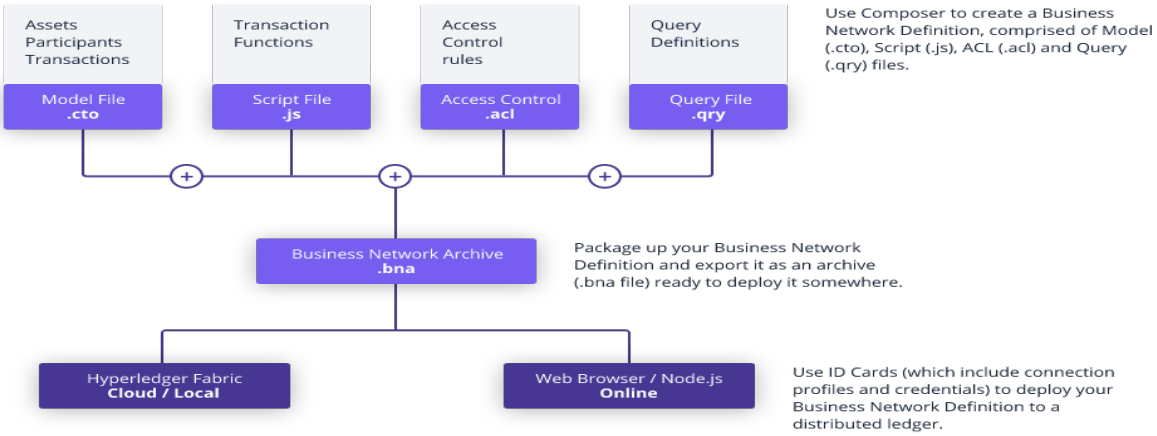
- Modelo de negocio: hace referencia al modelo de datos de la red compuesto por los activos, participantes, transacciones y eventos. Este modelo, que también define las relaciones y reglas de validación, se encuentra serializado en formato JSON (*JavaScript Object Notation*) y es validado en tiempo de ejecución.
- Procesador de transacciones: define en JS la lógica y requerimientos de negocio de cada transacción en funciones procesadoras. Este componente se considera el *chaincode* de HF ya que es el código que se despliega y ejecuta en cada nodo de la BC en tiempo de ejecución.
- Lista de control de acceso (*Access Control List* -ACL-): determina las reglas de quién tiene acceso y a qué recursos en tiempo de ejecución, es decir, establece la configuración de privacidad y compartición.

A mayores de posibilitar la definición de la red, HC ofrece:

- Composer Playground: herramienta web para probar una BND en una instancia de una red BC sin realizar ninguna instalación ni despliegue adicional lo que facilita el conocer si la funcionalidad definida es correcta o no antes de su puesta en producción.
- Soporte REST API y capacidades de integración con lo que la red desplegada puede ser fácilmente consumida por soluciones externas gracias a la API generada dinámicamente en el momento del despliegue.

La BND posee un nombre y una versión y cuando se quiere desplegar en la BC de HF como un *chaincode* genérico debe ser empaquetada en un fichero banana (`.bna`), tal y como muestra la Figura 2.9. La conexión con HF se realiza mediante los perfiles de conexión, documento JSON que especifica cómo se debe establecer la conexión con el sistema e incluye información como, direcciones TCP/IP y puertos para los *peers*, certificados criptográficos, etc.

Figura 2.9: Definición y despliegue de la red con Hyperledger Composer.

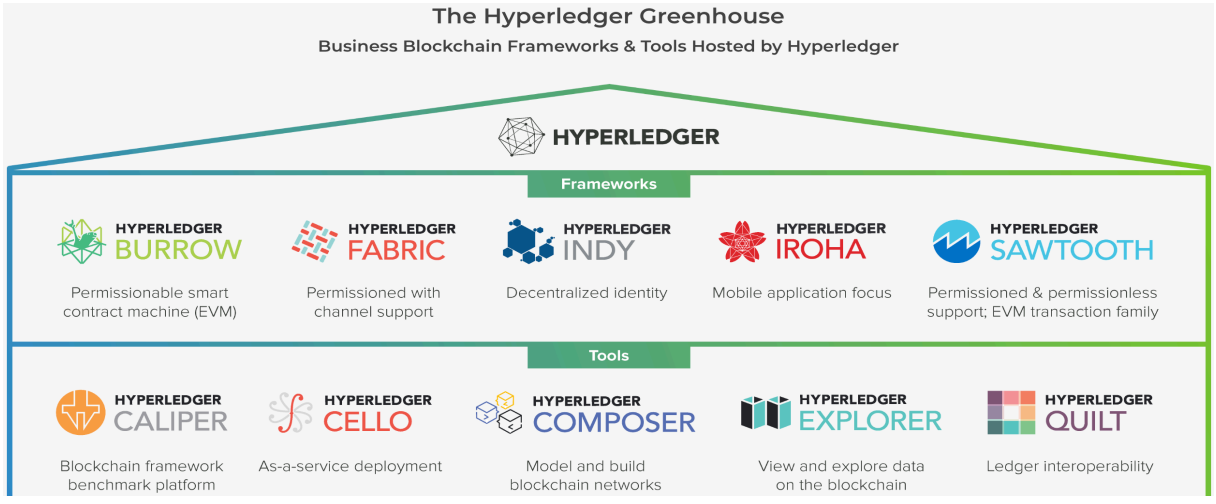


Fuente: `https://hyperledger.github.io/composer/latest/index.html`

2.6.3. Otro proyectos de Hyperledger

Además de HF y HC, el consorcio Hyperledger mantiene actualmente otros ocho marcos de trabajo (Figura 2.10), descritos brevemente a continuación:

Figura 2.10: Proyectos de la iniciativa Hyperledger.



Fuente: `https://www.hyperledger.org/`

1. Hyperledger Burrow: anteriormente Monax, fue impulsada por la *startup* Monax Industries y co-patrocinada por Intel. Es una BC privada basada en el código de Ethereum por lo

que permite la ejecución de *smart contracts* desarrollados en Solidity sobre la base de las especificaciones de la máquina virtual de esta tecnología (EVM).

2. Hyperledger Indy: plataforma cuyo principal objetivo es proponer una solución para la identidad descentralizada que apuesta por el protocolo criptográfico de conocimiento cero (*Zero Knowledge Proof*) el cual sienta sus bases en reducir la cantidad de información personal en procesos en los que se usa la identidad digital. Para ello, proporciona herramientas para crear y usar identidades digitales independientes que sean interoperables entre dominios y aplicaciones.

3. Hyperledger Iroha: *framework* desarrollado por Hitachi, NTT Data, entre otras compañías para incorporar a diversos proyectos, en especial a proyectos móviles, una plataforma BC simple, moderna y modularizada que permita el despliegue de *smart contracts* desarrollados en Java. Esta plataforma utilizada una variación del algoritmo de consenso tolerante a errores bizantinos (*Byzantine Fault Tolerant* -BFT-) llamado Sumeragi, conocido por su alto grado de tolerancia.

4. Hyperledger Sawtooth: BC privada de ámbito empresarial desarrollada mayormente por la compañía Intel. Está diseñado para la versatilidad y escalabilidad permitiendo crear redes altamente configurables. Dispone de un consenso propio denominado PoET (prueba de tiempo transcurrido -*Proof-of-Elapsed-Time*-) y la capacidad de ejecutar transacciones paralelas mediante *smart contracts* y *ledgers* verdaderamente distribuidos.

5. Hyperledger Caliper: herramienta que genera informes con diferentes indicadores de rendimiento como, por ejemplo el número de transacciones por segundo (TPS), la latencia de transacción, utilización de recursos, etc.

6. Hyperledger Cello: herramienta inicialmente aportada por IBM para ofrecer el modelo de implementación bajo demanda (*as-a-service*) al ecosistema de BC con la consiguiente reducción de esfuerzo a la hora crear, administrar y finalizar la infraestructura de red.

7. Hyperledger Explorer: herramienta inicialmente contribuida por IBM, Intel y DTCC que proporciona un sistema web para la consulta de la información de la red: listado de nodos, códigos de cadena, bloques, transacciones, etc.

8. Hyperledger Quilt: herramienta que ofrece interoperabilidad entre distintas redes BC mediante el protocolo ILP (*Interledger Protocol*). Este protocolo, creado por Ripple, es un protocolo de pagos que nació para solucionar la necesidad de transferir valor a través de *ledgers* distribuidos y no distribuidos.

Capítulo 3

Análisis

Conocer todos y cada uno de los requerimientos de un sistema es fundamental para no caminar a través del camino incorrecto. En este capítulo se aborda el primer contacto con el sistema a través de un completo análisis con el fin de indagar con mayor profundidad en él y de desglosar y definir las necesidades a implementar, todo ello bajo las directrices y pautas del marco de planificación detallado en el anexo B.

3.1. Requisitos

La especificación de requisitos es una característica esencial a la hora de analizar y diseñar un sistema ya que se puede considerar como la base de todo sistema. Se encarga de proporcionar una descripción del comportamiento y servicios del sistema que se va a desarrollar: funcionalidades que debe implementar, información que debe almacenar y producir, restricciones, etc. Resumidamente, definen las interacciones usuario-sistema y sistema-sistema.

Los requisitos, que deben ser establecidos durante la fase de elaboración del proyecto, pueden revisarse, modificarse e incluso añadir nuevos durante las iteraciones de la fase de construcción al contextualizar el proyecto en el marco de trabajo AUP (*Agile Unified Process*). Para el presente proyecto, en la primera iteración de la fase de construcción (iteración 3 de la planificación global) se realizó la revisión de los requerimientos junto a otros aspectos de la fase de análisis.

3.1.1. Requisitos funcionales (RF)[1]

Expresan y definen las funcionalidades que el sistema debe proporcionar y ejecutar, es decir, qué debe hacer el sistema.

[1]Los RF identificados hacen referencia al sistema en general sin especificar al componente en concreto al que pertenecen. En el capítulo 5 se detalla con profundidad la arquitectura y los componentes del proyecto.

¡Importante 1! - La especificación completa de cada requisito funcional contiene los campos especificados en el primer requisito. Para evitar alargar la memoria y la repetitividad, en los sucesivos requisitos funcionales se omiten aquellos campos de menor importancia y cuyo valor no es modificado.

¡Importante 2! - En el sistema web existen dos entidades que puede gestionar un usuario administrador: usuario administrador y usuario normal. La gestión de cada una de ellas engloba acciones idénticas y con el fin de evitar la repetitividad, solamente se detallará completamente los requisitos funcionales de la entidad Usuario administrador marcando con un asterisco en azul (*****) aquellos que sean similares para la otra entidad con la salvedad de su adaptación.

RF.1	
Nombre	**Instalar dependencias en el dispositivo Raspberry Pi (RPi)**
Versión	1.0 - 03/01/2018
Autores	Jesús Iglesias García
Dependencias	Ninguna
Descripción	El sistema permitirá *la instalación de paquetes y librerías Python en la RPi*
Importancia	Vital
Urgencia	Inmediata
Estado	En construcción
Estabilidad	Baja

Tabla 3.1: RF.1 - Instalar dependencias en el dispositivo Raspberry Pi (RPi)

RF.2	
Nombre	**Actualizar dependencias en el dispositivo RPi**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la actualización de paquetes y librerías Python en la RPi*

Tabla 3.2: RF.2 - Actualizar dependencias en el dispositivo RPi

RF.3	
Nombre	**Habilitar interfaces en el dispositivo RPi**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la activación de las interfaces I2C (Inter-Integrated Circuit) y camera de la RPi*

Tabla 3.3: RF.3 - Habilitar interfaces en el dispositivo RPi

RF.4	
Nombre	**Seleccionar acciones de configuración a ejecutar en el dispositivo RPi**
Versión	1.0 - 13/02/2018
Descripción	El sistema permitirá *la indicación de qué acciones ejecutar en la configuración de la RPi (solamente instalación y/o actualización de paquetes y librerías, solamente activación de interfaces o ambas)*

Tabla 3.4: RF.4 - Seleccionar acciones de configuración a ejecutar en el dispositivo RPi

RF.5	
Nombre	**Reiniciar dispositivo RPi**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *el reinicio de la RPi*

Tabla 3.5: RF.5 - Reiniciar dispositivo RPi

RF.6	
Nombre	**Tomar mediciones de sucesos del entorno**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la medición constante de determinados eventos del entorno para controlar si se produce algún suceso anómalo*

Tabla 3.6: RF.6 - Tomar mediciones de sucesos del entorno

RF.7	
Nombre	**Activar protocolo de alerta**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la activación de un protocolo de alerta en caso de que se produzca un suceso anómalo para la obtención de una evidencia que actuará como prueba de lo sucedido*

Tabla 3.7: RF.7 - Activar protocolo de alerta

RF.8	
Nombre	**Generar evidencia única**
Versión	1.1 - 03/01/2018 (rev. 13/02/2018)
Descripción	El sistema permitirá *la creación de una evidencia junto con una serie de propiedades que la identifican, entre ellas una fecha -timestamp- y un identificador único*

Tabla 3.8: RF.8 - Generar evidencia única

RF.9	
Nombre	**Capturar evidencia**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la grabación del entorno durante un tiempo establecido*

Tabla 3.9: RF.9 - Capturar evidencia

RF.10	
Nombre	**Encriptar y firmar evidencias**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la encriptación de evidencias para uno o varios destinatarios y el firmado de éstas con un solo destinatario*

Tabla 3.10: RF.10 - Encriptar y firmar evidencias

RF.11	
Nombre	**Calcular código *hash* de la evidencia sin encriptar**
Versión	1.1 - 03/01/2018 (rev. 13/02/2018)
Descripción	El sistema permitirá *el cálculo del valor hash del contenido de la evidencia el cual actuará como prueba de integridad*

Tabla 3.11: RF.11 - Calcular código hash de la evidencia sin encriptar

RF.12	
Nombre	**Añadir medición a la base de datos (BBDD)**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *el almacenamiento de mediciones en la BBDD*

Tabla 3.12: RF.12 - Añadir medición a la base de datos BBDD

RF.13	
Nombre	**Almacenar evidencia encriptada y firmada a la nube**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *el almacenamiento en la nube de la evidencia generada tras la activación del protocolo de alerta*

Tabla 3.13: RF.13 - Almacenar evidencia encriptada y firmada a la nube

RF.14	
Nombre	**Registrar evidencia en la Blockchain (BC)**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *el registro de una evidencia en la BC*

Tabla 3.14: RF.14 - Registrar evidencia en la Blockchain (BC)

RF.15	
Nombre	**Enviar *email* de notificación**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *el envío de emails para notificar un suceso anómalo (activación del protocolo de alerta) o un error producido durante una medición*

Tabla 3.15: RF.15 - Enviar email de notificación

RF.16	
Nombre	**Generar par de claves GPG (GNU Privacy Guard)**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la generación del par de claves GPG (clave privada y pública) para llevar a cabo la encriptación y firmado de la evidencia*

Tabla 3.16: RF.16 - Generar par de claves GPG (GNU Privacy Guard)

RF.17	
Nombre	**Configurar identidad del usuario para el par de claves GPG**
Versión	1.0 - 13/02/2018
Descripción	El sistema permitirá *la indicación del nombre y dirección email a asociar a la identidad del usuario del par de claves*

Tabla 3.17: RF.17 - Configurar identidad del usuario para el par de claves GPG

RF.18	
Nombre	**Exportar par de claves GPG**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la exportación de la clave privada y pública a un fichero con extensión* `.asc` *cuyo nombre será renombrado en caso de existir un fichero con el nombre por defecto*

Tabla 3.18: RF.18 - Exportar par de claves GPG

RF.19	
Nombre	**Generar código QR del par de claves GPG**
Versión	1.0 - 13/02/2018
Descripción	El sistema permitirá *la generación de un código QR a partir del par de claves creado*

Tabla 3.19: RF.19 - Generar código QR del par de claves GPG

RF.20	
Nombre	**Seleccionar par de claves para el firmado de la evidencia**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la indicación del par de claves a utilizar en el firmado de la evidencia cuando se especifican varios destinatarios para la encriptación*

Tabla 3.20: RF.20 - Seleccionar par de claves para el firmado de la evidencia

RF.21	
Nombre	**Añadir adjunto en el *email* de notificación**
Versión	1.0 - 13/02/2018
Descripción	El sistema permitirá *el anexo del fichero de log en el email de notificación*

Tabla 3.21: RF.21 - Añadir adjunto en el email de notificación

RF.22	
Nombre	**Establecer conexión con el servicio de BBDD**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la conexión con el servicio de BBDD ubicado en la nube*

Tabla 3.22: RF.22 - Establecer conexión con el servicio de BBDD

RF.23	
Nombre	**Introducir credenciales en el servicio de BBDD**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la indicación de las credenciales de usuario para establecer conexión con el servicio de BBDD*

Tabla 3.23: RF.23 - Introducir credenciales en el servicio de BBDD

RF.24	
Nombre	**Configurar nombre de la BBDD**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la configuración del nombre de la BBDD a emplear*

Tabla 3.24: RF.24 - Configurar nombre de la BBDD

RF.25	
Nombre	**Crear o abrir BBDD**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la creación -si no existe- o la apertura -si existe- de la BBDD para poder interactuar con ella*

Tabla 3.25: RF.25 - Crear o abrir BBDD

RF.26	
Nombre	**Establecer conexión con el servicio de almacenamiento**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la conexión con el servicio de almacenamiento ubicado en la nube*

Tabla 3.26: RF.26 - Establecer conexión con el servicio de almacenamiento

RF.27	
Nombre	**Introducir credenciales en el servicio de almacenamiento en la nube**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la indicación del token de autenticación para establecer conexión con el servicio de almacenamiento*

Tabla 3.27: RF.27 - Introducir credenciales en el servicio de almacenamiento en la nube

RF.28	
Nombre	**Configurar nombre de los subdirectorios en el servicio de almacenamiento**
Versión	1.1 - 03/01/2018 (rev. 13/02/2018)
Descripción	El sistema permitirá *la configuración del nombre de los subdirectorios asociados a cada sensor donde se almacenarán las evidencias generadas por alertas lanzadas por dicho sensor*

Tabla 3.28: RF.28 - Configurar nombre de los subdirectorios en el servicio de almacenamiento

RF.29	
Nombre	**Crear subdirectorios en el servicio de almacenamiento**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la creación -si no existen- de los subdirectorios asociados a cada sensor*

Tabla 3.29: RF.29 - Crear subdirectorios en el servicio de almacenamiento

RF.30	
Nombre	**Comprobar espacio disponible en el servicio de almacenamiento**
Versión	1.0 - 13/02/2018
Descripción	El sistema permitirá *la comprobación del espacio disponible en el servicio de almacenamiento con el fin de notificar al usuario si el espacio restante supera una cantidad mínima recomendada*

Tabla 3.30: RF.30 - Comprobar espacio disponible en el servicio de almacenamiento

RF.31	
Nombre	**Obtener enlace de la evidencia**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la obtención del enlace de la evidencia almacenada en la nube*

Tabla 3.31: RF.31 - Obtener enlace de la evidencia

RF.32	
Nombre	**Comprobar disponibilidad del servidor que expone la red de negocio**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la comprobación de la disponibilidad del servidor que expone la red de negocio desplegada en la BC mediante una llamada ping*

Tabla 3.32: RF.32 - Comprobar disponibilidad del servidor que expone la red de negocio

RF.33	
Nombre	**Introducir dirección y puerto del servidor que expone la red de negocio y poseedor de las evidencias registradas**
Versión	1.1 - 03/01/2018 (rev. 13/02/2018)
Descripción	El sistema permitirá *la indicación de la URL (Uniform Resource Locator) y el puerto donde el servidor se encuentra desplegado y la información del poseedor de la evidencia -alerta- registrada, creando además un participante en la BC con esta información*

Tabla 3.33: RF.33 - Introducir dirección y puerto del servidor que expone la red de negocio y poseedor de las evidencias registradas

RF.34	
Nombre	**Securizar servidor que expone la red de negocio**
Versión	1.0 - 13/02/2018
Descripción	El sistema permitirá *la securización del servidor que expone la red de negocio mediante un token o api-key de acceso de tal forma que toda petición que no lo incluya será denegada*

Tabla 3.34: RF.34 - Securizar servidor que expone la red de negocio

RF.35	
Nombre	**Seleccionar dirección *email* donde enviar las notificaciones**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la indicación de la dirección email destino donde enviar notificaciones de alerta o de error durante una medición*

Tabla 3.35: RF.35 - Seleccionar dirección email donde enviar las notificaciones

RF.36	
Nombre	**Seleccionar rango máximo del evento distancia**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la indicación de la distancia máxima a ser medida por el sensor HC-SR04*

Tabla 3.36: RF.36 - Seleccionar rango máximo del evento distancia

RF.37	
Nombre	**Seleccionar tiempo de grabación**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la indicación del tiempo de grabación del entorno cuando el protocolo de alerta es activado*

Tabla 3.37: RF.37 - Seleccionar tiempo de grabación

RF.38	
Nombre	**Seleccionar frecuencia de medición**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la indicación del tiempo de espera entre una medición de sucesos y la siguiente*

Tabla 3.38: RF.38 - Seleccionar frecuencia de medición de sucesos

RF.39	
Nombre	**Seleccionar pin de datos del sensor DHT-11**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la indicación del pin de datos a utilizar en el sensor DHT-11*

Tabla 3.39: RF.39 - Seleccionar pin de datos del sensor DHT-11

RF.40	
Nombre	**Seleccionar pin *Echo* del sensor HC-SR04**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la indicación del pin Echo a utilizar en el sensor HC-SR04*

Tabla 3.40: RF.40 - Seleccionar pin Echo del sensor HC-SR04

RF.41	
Nombre	**Seleccionar pin *Trigger* del sensor HC-SR04**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la indicación del pin Trigger a utilizar en el sensor HC-SR04*

Tabla 3.41: RF.41 - Seleccionar pin Trigger del sensor HC-SR04

RF.42	
Nombre	**Seleccionar pin para el elemento LED (*Light-Emitting Diode*)**
Versión	1.0 - 03/01/2018
Descripción	El sistema permitirá *la indicación del pin a utilizar para el diodo emisor de luz (LED) de color rojo*

Tabla 3.42: RF.42 - Seleccionar pin para el elemento LED (Light-Emitting Diode)

RF.43	
Nombre	**Seleccionar tipo de dispositivo I2C para el LCD (*Liquid Crystal Display*) del sensor DHT-11**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la indicación del tipo de dispositivo I2C -entre un rango predefinido- para el LCD del sensor DHT-11*

Tabla 3.43: RF.43 - Seleccionar tipo de dispositivo I2C para el LCD (Liquid Crystal Display) del sensor DHT-11

RF.44	
Nombre	**Dirección I2C del LCD enlazado al sensor DHT-11**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la introducción de la dirección I2C que posee el LCD enlazado al sensor DHT-11*

Tabla 3.44: RF.44 - Dirección I2C del LCD enlazado al sensor DHT-11

RF.45	
Nombre	**Seleccionar tipo de dispositivo I2C para el LCD del sensor HC-SR04**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la indicación del tipo de dispositivo I2C -entre un rango predefinido- para el LCD del sensor HC-SR04*

Tabla 3.45: RF.45 - Seleccionar tipo de dispositivo I2C para el LCD del sensor HC-SR04

RF.46	
Nombre	**Dirección I2C del LCD enlazado al sensor HC-SR04**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la introducción de la dirección I2C que posee el LCD enlazado al sensor HC-SR04*

Tabla 3.46: RF.46 - Dirección I2C del LCD enlazado al sensor HC-SR04

RF.47	
Nombre	**Seleccionar umbral del evento humedad**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la indicación del umbral límite a partir del cual se activa el protocolo de alerta para el evento humedad*

Tabla 3.47: RF.47 - Seleccionar umbral del evento humedad

RF.48	
Nombre	**Seleccionar umbral del evento temperatura**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la indicación del umbral límite a partir del cual se activa el protocolo de alerta para el evento temperatura*

Tabla 3.48: RF.48 - Seleccionar umbral del evento temperatura

RF.49	
Nombre	**Seleccionar umbral del evento distancia**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la indicación del umbral límite a partir del cual se activa el protocolo de alerta para el evento distancia*

Tabla 3.49: RF.49 - Seleccionar umbral del evento distancia

RF.50	
Nombre	**Mostrar información de valores establecidos**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la visualización de algunos parámetros establecidos para la monitorización de sucesos del entorno antes de que ésta comience*

Tabla 3.50: RF.50 - Mostrar información de valores establecidos

RF.51	
Nombre	**Fichero de *log***
Versión	1.0 - 13/02/2018
Descripción	El sistema permitirá *la redirección de la salida del terminal a un fichero de log el cual será generado en tiempo de ejecución junto con el directorio donde se almacenará*

Tabla 3.51: RF.51 - Fichero de log

RF.52	
Nombre	**Desencriptar evidencia**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la desencriptación de la evidencia previamente encriptada y firmada con GPG*

Tabla 3.52: RF.52 - Desencriptar evidencia

RF.53	
Nombre	**Verificar firma de la evidencia**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la verificación de la identidad del usuario que firmó la evidencia*

Tabla 3.53: RF.53 - Verificar firma de la evidencia

RF.54	
Nombre	**Comparar valores *hash***
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la comparación de valores hash para verificar la integridad del contenido de la evidencia una vez desencriptada*

Tabla 3.54: RF.54 - Comparar valores hash

RF.55	
Nombre	**Seleccionar método para indicar el par de claves**
Versión	1.1 - 04/01/2018 (rev. 13/02/2018)
Descripción	El sistema permitirá *la selección de un método para indicar el par de claves a utilizar (importación o indicación de la huella digital -fingerprint-)*

Tabla 3.55: RF.55 - Seleccionar método para indicar el par de claves

RF.56	
Nombre	**Importar par de claves**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la importación del par de claves (pública y privada) desde un fichero para efectuar la desencriptación de la evidencia y la verificación de la firma*

Tabla 3.56: RF.56 - Importar par de claves

RF.57	
Nombre	**Introducir huella digital**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la indicación de la huella digital (fingerprint) asociada a una clave existente a usar para efectuar la encriptación/desencriptación de la evidencia y el firmado y verificación*

Tabla 3.57: RF.57 - Introducir huella digital

RF.58	
Nombre	**Introducir contraseña de la clave privada**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la indicación de la contraseña (passphrase) de protección de la clave privada para poder efectuar la desencriptación y firmado de la evidencia*

Tabla 3.58: RF.58 - Introducir contraseña de la clave privada

RF.59	
Nombre	**Seleccionar evidencia**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la indicación de la evidencia a desencriptar y verificar la firma*

Tabla 3.59: RF.59 - Seleccionar evidencia

RF.60	
Nombre	**Descargar evidencia**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la descarga de la evidencia almacenada en la nube*

Tabla 3.60: RF.60 - Descargar evidencia

RF.61	
Nombre	**Seleccionar método para indicar la evidencia**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la selección de un método para indicar la evidencia a utilizar (evidencia en el sistema local o descarga)*

Tabla 3.61: RF.61 - Seleccionar método para indicar la evidencia

RF.62	
Nombre	**Seleccionar directorio destino para la desencriptación**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la indicación del directorio donde la evidencia desencriptada será almacenada*

Tabla 3.62: RF.62 - Seleccionar directorio destino para la desencriptación

RF.63	
Nombre	**Introducir valor *hash***
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la indicación de valor hash para poder efectuar la verificación de integridad de la evidencia desencriptada*

Tabla 3.63: RF.63 - Introducir valor hash

RF.64	
Nombre	**Seleccionar directorio GPG**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la selección del directorio donde GPG generará su almacén*

Tabla 3.64: RF.64 - Seleccionar directorio GPG

RF.65	
Nombre	**Mostrar ayuda**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la visualización de la ayuda del componente*

Tabla 3.65: RF.65 - Mostrar ayuda

RF.66	
Nombre	**Mostrar información sobre la ejecución por consola**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la visualización en el terminal de las acciones ejecutadas*

Tabla 3.66: RF.66 - Mostrar información sobre la ejecución por consola

RF.67	
Nombre	**Mostrar más información sobre la ejecución**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *en algunos componentes mostrar más información sobre las acciones ejecutadas*

Tabla 3.67: RF.67 - Mostrar más información sobre la ejecución

RF.68	
Nombre	**Mostrar información a través de los LCDs**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la visualización de determinada información sobre la ejecución por medio de los LCDs enlazados a los sensores*

Tabla 3.68: RF.68 - Mostrar información a través de los LCDs

RF.69	
Nombre	**Activar LED**
Versión	1.0 - 13/02/2018
Descripción	El sistema permitirá *la activación del LED rojo como señal visual de que un suceso anómalo se ha producido y se ha activado el protocolo de alerta*

Tabla 3.69: RF.69 - Activar LED

RF.70	
Nombre	**Valores por defecto**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *usar valores por defecto pre-configurados en caso de no parametrizar la ejecución*

Tabla 3.70: RF.70 - Valores por defecto

RF.71	
Nombre	**Inicializar componentes y servicios**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la inicialización de los componentes hardware, directorios locales, utilidades y servicios a utilizar para un correcto funcionamiento posterior*

Tabla 3.71: RF.71 - Inicializar componentes y servicios

RF.72	
Nombre	**Finalizar ordenadamente componentes y servicios**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la finalización ordenada de los componentes hardware, utilidades y servicios a utilizar mediante la liberación de las instancias generadas*

Tabla 3.72: RF.72 - Finalizar ordenadamente componentes y servicios

RF.73	
Nombre	**Gestionar errores**
Versión	1.0 - 04/01/2018
Descripción	El sistema permitirá *la gestión de cualquier error que se produzca durante la ejecución informando al usuario de tal suceso*

Tabla 3.73: RF.73 - Gestionar errores

RF.74	
Nombre	**Identificación de usuario**
Versión	1.0 - 04/01/2018
Descripción	El sistema web permitirá *iniciar sesión a un usuario (normal o administrador) cuando se haya autentificado correctamente a través de la introducción de las credenciales*

Tabla 3.74: RF.74 - Identificación de usuario

RF.75	
Nombre	**Activación de la opción Recordarme**
Versión	1.0 - 04/01/2018
Descripción	El sistema web permitirá *al usuario (normal o administrador) activar la autenticación automática*

Tabla 3.75: RF.75 - Activación de la opción Recordarme

RF.76	
Nombre	**Autenticación a través de la opción Recordarme**
Versión	1.0 - 04/01/2018
Descripción	El sistema web permitirá *la autenticación automática cuando esté seleccionada la opción Recordarme*

Tabla 3.76: RF.76 - Autenticación a través de la opción Recordarme

RF.77	
Nombre	**Información cuando el estado de cuenta deniegue el acceso al iniciar sesión**
Versión	1.0 - 04/01/2018
Descripción	El sistema web permitirá *al usuario conocer la información del estado de la cuenta cuando el sistema no le permita el acceso al intentar iniciar sesión*

Tabla 3.77: RF.77 - Información cuando el estado de cuenta deniegue el acceso al iniciar sesión

RF.78	
Nombre	**Restablecer contraseña**
Versión	1.0 - 04/01/2018
Descripción	El sistema web permitirá *a un usuario (normal o administrador) restablecer su contraseña*

Tabla 3.78: RF.78 - Restablecer contraseña

RF.79	
Nombre	**Búsqueda rápida de usuario normal**
Versión	1.0 - 04/01/2018
Descripción	El sistema web permitirá *al usuario administrador realizar una búsqueda rápida de un usuario normal*

Tabla 3.79: RF.79 - Búsqueda rápida de usuario normal

RF.80	
Nombre	**Gestionar usuario administrador ***
Versión	1.0 - 04/01/2018
Descripción	El sistema web permitirá *al usuario administrador realizar la gestión de usuarios administradores*

Tabla 3.80: RF.80 - Gestionar usuario administrador

RF.81	
Nombre	**Consultar lista de usuarios administradores ***
Versión	1.0 - 04/01/2018
Descripción	El sistema web permitirá *al usuario administrador consultar la lista de usuarios administradores registrados en el sistema*

Tabla 3.81: RF.81 - Consultar lista de usuarios administradores

RF.82	
Nombre	**Buscar usuario administrador ***
Versión	1.0 - 04/01/2018
Descripción	El sistema web permitirá *al usuario administrador realizar una búsqueda sobre el listado de usuarios administradores existentes en el sistema*

Tabla 3.82: RF.82 - Buscar usuario administrador

RF.83	
Nombre	**Ordenar usuarios administradores ***
Versión	1.0 - 13/02/2018
Descripción	El sistema web permitirá *al usuario administrador ordenar de manera ascendente o descendente los usuarios administradores según el campo seleccionado*

Tabla 3.83: RF.83 - Ordenación de usuarios administradores

RF.84	
Nombre	**Filtrar usuarios administradores ***
Versión	1.0 - 13/02/2018
Descripción	El sistema web permitirá *al usuario administrador filtrar el número de usuarios administradores a mostrar*

Tabla 3.84: RF.84 - Filtrar usuarios administradores

RF.85	
Nombre	**Visualizar y restablecer campos en la lista de usuarios administradores ***
Versión	1.0 - 13/02/2018
Descripción	El sistema web permitirá *al usuario administrador visualizar en el listado de los usuarios administradores los campos deseados de éste a través de la activación de las columnas correspondientes y restablecer la visualización de los campos marcados como principales*

Tabla 3.85: RF.85 - Visualizar y restablecer campos en la lista de usuarios administradores

RF.86	
Nombre	**Exportar usuarios administradores ***
Versión	1.0 - 13/02/2018
Descripción	El sistema web permitirá *al usuario administrador exportar en formato PDF y CSV los usuarios administradores del sistema*

Tabla 3.86: RF.86 - Exportar usuarios administradores

RF.87	
Nombre	**Imprimir usuarios administradores** *
Versión	1.0 - 13/02/2018
Descripción	El sistema web permitirá *al usuario administrador imprimir el listado de usuarios administradores del sistema*

Tabla 3.87: RF.87 - Imprimir usuarios administradores

RF.88	
Nombre	**Copiar usuarios administradores** *
Versión	1.0 - 04/01/2018
Descripción	El sistema web permitirá *al usuario administrador copiar al portapapeles el listado de usuarios administradores del sistema*

Tabla 3.88: RF.88 - Copiar usuarios administradores

RF.89	
Nombre	**Crear usuario administrador** *
Versión	1.0 - 04/01/2018
Descripción	El sistema web permitirá *al usuario administrador la creación de nuevos administradores en el sistema*

Tabla 3.89: RF.89 - Crear usuario administrador

RF.90	
Nombre	**Editar usuario administrador** *
Versión	1.0 - 04/01/2018
Descripción	El sistema web permitirá *al usuario administrador editar los usuarios administradores existentes en el sistema*

Tabla 3.90: RF.90 - Editar usuario administrador

RF.91	
Nombre	**Eliminar cuenta de usuario administrador ***
Versión	1.0 - 04/01/2018
Descripción	El sistema web permitirá *al usuario administrador eliminar una cuenta de usuario administrador*

Tabla 3.91: RF.91 - Eliminar cuenta de usuario administrador

RF.92	
Nombre	**Bloquear, inhabilitar, expirar cuenta o contraseña de usuario administrador ***
Versión	1.0 - 04/01/2018
Descripción	El sistema web permitirá *al usuario administrador bloquear, inhabilitar, expirar cualquier cuenta o contraseña de un usuario administrador*

Tabla 3.92: RF.92 - Bloquear, inhabilitar, expirar cuenta o contraseña de usuario administrador

RF.93	
Nombre	**Activar cuenta de usuario administrador ***
Versión	1.0 - 04/01/2018
Descripción	El sistema web permitirá *al usuario administrador activar cualquier cuenta de un usuario administrador previamente bloqueada, inactiva o expirada*

Tabla 3.93: RF.93 - Activar cuenta de usuario administrador

RF.94	
Nombre	**Consultar estadísticas**
Versión	1.0 - 04/01/2018
Descripción	El sistema web permitirá *al usuario administrador visualizar estadísticas globales y al usuario normal visualizar sus propias estadísticas*

Tabla 3.94: RF.94 - Consultar estadísticas

RF.95	
Nombre	**Consultar perfil personal**
Versión	1.0 - 04/01/2018
Descripción	El sistema web permitirá *al usuario normal consultar los datos de su perfil personal*

Tabla 3.95: RF.95 - Consultar perfil personal

RF.96	
Nombre	**Editar perfil personal**
Versión	1.1 - 04/01/2018 (rev. 13/02/2018)
Descripción	El sistema web permitirá *al usuario normal editar los datos de su perfil personal*

Tabla 3.96: RF.96 - Editar perfil personal

RF.97	
Nombre	**Visualizar información sobre las mediciones y sucesos anómalos**
Versión	1.0 - 04/01/2018
Descripción	El sistema web permitirá *al usuario administrador visualizar la información completa y filtrada en tiempo real sobre las mediciones tomadas y los sucesos anómalos (alertas) producidos. Además, permitirá al usuario normal visualizar la información monitorizada de la que sea poseedor*

Tabla 3.97: RF.97 - Visualizar información sobre las mediciones y sucesos anómalos

3.1.2. Requisitos no funcionales (RNF)

También denominados cualidades del sistema, especifican las restricciones que afectan a la funcionalidad del sistema, es decir, cómo el sistema debe ser y comportarse.

¡Importante! - Al igual que en los requisitos funcionales, en los requisitos no funcionales solamente se especificarán los campos de interés con la salvedad del primero que contiene la especificación completa.

RNF.1	
Categoría	**Facilidad de uso**
Nombre	**Interfaz de usuario (*User Interface* -UI-)**
Versión	1.0 - 05/01/2018
Autores	Jesús Iglesias García
Dependencias	Ninguna
Descripción	El sistema deberá *ofrecer una UI fácil de usar e intuitiva para los usuarios finales*
Importancia	Vital
Urgencia	Inmediata
Estado	En construcción
Estabilidad	Baja

Tabla 3.98: RNF.1 - Interfaz de usuario (User Interface -UI-)

RNF.2	
Categoría	**Facilidad de uso**
Nombre	**Contenido descriptivo**
Versión	1.0 - 05/01/2018
Descripción	El sistema deberá *contener acciones, iconos y textos descriptivos y significativos para los usuarios finales*

Tabla 3.99: RNF.2 - Contenido descriptivo

RNF.3	
Categoría	**Facilidad de uso**
Nombre	**Interactivo**
Versión	1.0 - 05/01/2018
Descripción	El sistema deberá *ser interactivo y contener un flujo de información lógico para los usuarios finales*

Tabla 3.100: RNF.3 - Interactivo

RNF.4	
Categoría	**Facilidad de uso**
Nombre	**Consistencia**
Versión	1.0 - 05/01/2018
Descripción	El sistema deberá *ser consistente para los usuarios finales respecto a la apariencia, disposición y comportamiento de los elementos*

Tabla 3.101: RNF.4 - Consistencia

RNF.5	
Categoría	**Facilidad de uso**
Nombre	**Diseño adaptativo**
Versión	1.0 - 05/01/2018
Descripción	El sistema web deberá *poseer un diseño responsive, es decir, visualizable correctamente en distintos navegadores y tipos de dispositivos así como en diferentes tamaños*

Tabla 3.102: RNF.5 - Diseño adaptativo

RNF.6	
Categoría	**Rendimiento**
Nombre	**Tiempo de autenticación**
Versión	1.0 - 05/01/2018
Descripción	La autenticación de usuario y de los servicios *se realizará mediante credenciales o token de acceso y con una duración menor a 10 segundos*

Tabla 3.103: RNF.6 - Tiempo de autenticación

RNF.7	
Categoría	**Rendimiento**
Nombre	**Tiempo de solicitud**
Versión	1.0 - 05/01/2018
Descripción	El sistema expondrá *la información solicitada por el usuario en un máximo de 10 segundos*

Tabla 3.104: RNF.7 - Tiempo de solicitud

RNF.8	
Categoría	**Rendimiento**
Nombre	**Tiempo de ejecución**
Versión	1.0 - 05/01/2018
Descripción	El sistema ejecutará *cualquier acción interna en un tiempo máximo de 1 minuto siempre y cuando no dependa de un factor externo (p.ej. velocidad de red)*

Tabla 3.105: RNF.8 - Tiempo de ejecución

RNF.9	
Categoría	**Rendimiento**
Nombre	**Tiempo de cierre de sesión**
Versión	1.0 - 05/01/2018
Descripción	El sistema web cerrará *la sesión del usuario transcurrido el tiempo máximo establecido (por defecto 30 minutos)*

Tabla 3.106: RNF.9 - Tiempo de cierre de sesión

RNF.10	
Categoría	**Rendimiento**
Nombre	**Modo de funcionamiento**
Versión	1.0 - 05/01/2018
Descripción	El sistema deberá *funcionar de forma fiable e ininterrumpida desde su ejecución hasta que sea parado explícitamente*

Tabla 3.107: RNF.10 - Modo de funcionamiento

RNF.11	
Categoría	**Rendimiento**
Nombre	**Reiniciar dispositivo**
Versión	1.0 - 05/01/2018
Descripción	El sistema deberá *reiniciar el dispositivo RPi en un periodo de 5 segundos como máximo*

Tabla 3.108: RNF.11 - Reiniciar dispositivo

RNF.12	
Categoría	**Rendimiento**
Nombre	**Prestaciones**
Versión	1.0 - 05/01/2018
Descripción	El sistema presentará *la menor carga posible en cuanto a consumo de almacenamiento y memoria*

Tabla 3.109: RNF.12 - Prestaciones

RNF.13	
Categoría	**Restricciones de diseño**
Nombre	**Tecnologías de desarrollo**
Versión	1.0 - 05/01/2018
Descripción	Se emplearán *las tecnologías: Python, Bash (Bourne-again Shell), Grails, JavaScript (JS), Groovy, Bootstrap, jQuery, HTML (HyperText Markup Language) y CSS (Cascading Style Sheets) para el desarrollo de los diferentes componentes del sistema*

Tabla 3.110: RNF.13 - Tecnologías de desarrollo

RNF.14	
Categoría	**Restricciones de diseño**
Nombre	**Patrones de diseño**
Versión	1.0 - 05/01/2018
Descripción	Se emplearán *diversos patrones de diseño para la codificación del sistema*

Tabla 3.111: RNF.14 - Patrones de diseño

RNF.15	
Categoría	**Restricciones de diseño**
Nombre	**Servicios en la nube**
Versión	1.0 - 05/01/2018
Descripción	Se emplearán *diversos servicios localizados en la nube para la persistencia del sistema (p.ej. servicio de almacenamiento, BBDD, etc.)*

Tabla 3.112: RNF.15 - Servicios en la nube

RNF.16	
Categoría	**Restricciones de diseño**
Nombre	**Hyperledger Fabric (HF)**
Versión	1.0 - 05/01/2018
Descripción	Se empleará *HF como tecnología de BC*

Tabla 3.113: RNF.16 - Hyperledger Fabric (HF)

RNF.17	
Categoría	**Restricciones de diseño**
Nombre	**Prototipo *hardware***
Versión	1.0 - 05/01/2018
Descripción	Se empleará *un dispositivo RPi, los sensores DHT-11 y HC-SR04, dos LCDs y un LED para construir el prototipo*

Tabla 3.114: RNF.17 - Prototipo hardware

RNF.18	
Categoría	**Seguridad**
Nombre	**Autenticación**
Versión	1.0 - 05/01/2018
Descripción	El sistema web realizará *un control de seguridad para la autenticación del usuario que se realizará mediante un identificador de usuario/email y contraseña*

Tabla 3.115: RNF.18 - Autenticación

RNF.19	
Categoría	**Seguridad**
Nombre	**Encriptar contraseñas**
Versión	1.0 - 05/01/2018
Descripción	El sistema web *deberá encriptar las contraseñas*

Tabla 3.116: RNF.19 - Encriptar contraseñas

RNF.20	
Categoría	**Seguridad**
Nombre	**Protocolo de comunicación**
Versión	1.0 - 05/01/2018
Descripción	Se empleará *en la medida de lo posible el protocolo HTTPS (Hypertext Transfer Protocol) para securizar la comunicación con los componentes y servicios*

Tabla 3.117: RNF.20 - Protocolo de comunicación

RNF.21	
Categoría	**Fiabilidad**
Nombre	**Integridad de datos**
Versión	1.0 - 05/01/2018
Descripción	El sistema web deberá *asegurar la integridad de los datos de los usuarios registrados*

Tabla 3.118: RNF.21 - Integridad de datos

RNF.22	
Categoría	**Funcionalidad del sistema**
Nombre	**Usuario superprivilegiado**
Versión	1.0 - 05/01/2018
Descripción	El sistema solamente iniciará *determinados componentes si el usuario que lo ejecuta es superprivilegiado*

Tabla 3.119: RNF.22 - Usuario superprivilegiado

RNF.23	
Categoría	**Funcionalidad del sistema**
Nombre	**Ejecución en plataforma GNU/Linux**
Versión	1.0 - 05/01/2018
Descripción	Algunos componentes del sistema deberán *ejecutarse en una plataforma GNU/Linux*

Tabla 3.120: RNF.23 - Ejecución en plataforma GNU/Linux

RNF.24	
Categoría	**Funcionalidad del sistema**
Nombre	**Ejecución en RPi**
Versión	1.0 - 05/01/2018
Descripción	Algunos componentes del sistema deberán *ejecutarse en una RPi*

Tabla 3.121: RNF.24 - Ejecución en RPi

RNF.25	
Categoría	**Funcionalidad del sistema**
Nombre	**Conexión a Internet**
Versión	1.0 - 05/01/2018
Descripción	Algunos componentes del sistema deberán *poseer conexión a la red para su ejecución*

Tabla 3.122: RNF.25 - Conexión a Internet

RNF.26	
Categoría	**Funcionalidad del sistema**
Nombre	**Ejecución o acceso concurrente**
Versión	1.0 - 05/01/2018
Descripción	El sistema limitará *la ejecución de algunos componentes a solamente una instancia simultánea. Además, limitará el acceso al sistema web a 1 acceso/sesión para los usuarios normales. Los usuarios administradores no presentan esta limitación*

Tabla 3.123: RNF.26 - Ejecución o acceso concurrente

RNF.27	
Categoría	**Funcionalidad del sistema**
Nombre	**Datos de entrada**
Versión	1.0 - 05/01/2018
Descripción	El sistema deberá *restringir el formato de los datos de entrada (tipo, valores, patrón de contraseña, etc.)*

Tabla 3.124: RNF.27 - Datos de entrada

RNF.28	
Categoría	**Funcionalidad del sistema**
Nombre	**Datos iniciales**
Versión	1.1 - 05/01/2018 (rev. 13/02/2018)
Descripción	El sistema web deberá *permitir cargar datos iniciales durante su arranque*

Tabla 3.125: RNF.28 - Datos iniciales

RNF.29	
Categoría	**Funcionalidad del sistema**
Nombre	**Crear usuario normal**
Versión	1.1 - 05/01/2018 (rev. 13/02/2018)
Descripción	El sistema web deberá *permitir crear un usuario normal solamente cuando su nombre de usuario esté disponible en el sistema web y exista un participante con el mismo nombre de usuario en la BC*

Tabla 3.126: RNF.29 - Crear usuario normal

RNF.30	
Categoría	**Funcionalidad del sistema**
Nombre	**Ocultar información**
Versión	1.0 - 05/01/2018
Descripción	El sistema web deberá *ocultar cierta información según el usuario o rol de éste*

Tabla 3.127: RNF.30 - Ocultar información

RNF.31	
Categoría	**Funcionalidad del sistema**
Nombre	**Redirección usuario administrador**
Versión	1.0 - 05/01/2018
Descripción	El sistema web deberá *redirigir al usuario administrador al panel de control tras autenticarse*

Tabla 3.128: RNF.31 - Redirección usuario administrador

RNF.32	
Categoría	**Funcionalidad del sistema**
Nombre	**Redirección usuario normal**
Versión	1.0 - 05/01/2018
Descripción	El sistema web deberá *redirigir al usuario normal a la página de usuario tras autenticarse*

Tabla 3.129: RNF.32 - Redirección usuario normal

RNF.33	
Categoría	**Funcionalidad del sistema**
Nombre	**Restricción de acceso**
Versión	1.0 - 05/01/2018
Descripción	El sistema web deberá *restringir el acceso a las distintas secciones según el rol de usuario*

Tabla 3.130: RNF.33 - Restricción de acceso

RNF.34	
Categoría	**Funcionalidad del sistema**
Nombre	**Restricción de acceso de usuario sin registrar**
Versión	1.0 - 05/01/2018
Descripción	El sistema web deberá *restringir el acceso a usuarios no registrados*

Tabla 3.131: RNF.34 - Restricción de acceso de usuario sin registrar

RNF.35	
Categoría	**Funcionalidad del sistema**
Nombre	**Restricción de acceso de usuario inactivo y bloqueado o con cuenta o contraseña expirada**
Versión	1.0 - 05/01/2018
Descripción	El sistema web deberá *restringir el acceso al sistema a un usuario cuya cuenta haya sido bloqueada, se encuentra expirada o la contraseña expiró*

Tabla 3.132: RNF.35 - Restricción de acceso de usuario inactivo y bloqueado o con cuenta o contraseña expirada

RNF.36	
Categoría	**Funcionalidad del sistema**
Nombre	**Limitación de validez temporal del *token* de restablecimiento de contraseña**
Versión	1.1 - 05/01/2018 (rev. 13/02/2018)
Descripción	El sistema web limitará *la validez temporal (30 minutos) del token de restablecimiento de contraseña*

Tabla 3.133: RNF.36 - Limitación de validez temporal del token de restablecimiento de contraseña

RNF.37	
Categoría	**Funcionalidad del sistema**
Nombre	**Limitación de visualización de información en el sistema web sobre la monitorización**
Versión	1.0 - 05/01/2018
Descripción	El sistema limitará *a los usuarios normales del sistema web a visualizar únicamente su información monitorizada, es decir, mediciones y sucesos anómalos o alertas de los que sean poseedores*

Tabla 3.134: RNF.37 - Limitación de visualización de información en el sistema web sobre la monitorización

RNF.38	
Categoría	**Funcionalidad del sistema**
Nombre	**Limitación de acceso a la red de negocio desplegada en la BC**
Versión	1.0 - 05/01/2018
Descripción	El sistema definirá *una lista de control de acceso sobre la red de negocio con la que restringe determinadas acciones según el tipo de participante*

Tabla 3.135: RNF.38 - Limitación de acceso a la red de negocio desplegada en la BC

3.1.3. Requisitos de información

Recopilan la información que debe persistir el sistema. Solamente se especifican las entidades principales -la primera de forma completa-, omitiendo aquellas que son secundarias o son establecidas por *plugins* como, por ejemplo la relación usuario-rol en el sistema web, etc.

RI.1	
Nombre	**Medición**
Versión	1.0 - 08/01/2018
Autores	Jesús Iglesias García
Dependencias	Ninguna
Descripción	El sistema debe *almacenar la siguiente información sobre cada medición:* ▪ *Identificador de la medición* ▪ *Dirección email del destinatario para efectos de notificación* ▪ *Evento del sensor que origina el protocolo de alerta* ▪ *Marca temporal (timestamp)* ▪ *Link de la evidencia* ▪ *Protocolo de alerta activado o no* ▪ *Sensor que origina el protocolo de alerta* ▪ *Umbral límite del evento que lanza el protocolo de alerta* ▪ *Usuario poseedor de la alerta* ▪ *Valor del evento temperatura* ▪ *Valor del evento humedad* ▪ *Valor del evento distancia*
Importancia	Vital
Urgencia	Inmediata
Estado	En construcción
Estabilidad	Baja

Tabla 3.136: RI.1 - Medición

RI.2	
Nombre	**Evidencia**
Versión	1.0 - 08/01/2018
Descripción	El sistema debe *almacenar la siguiente información sobre cada evidencia:* ▪ *Fichero de evidencia encriptado y firmado*

Tabla 3.137: RI.2 - Evidencia

RI.3	
Nombre	**Alerta en BC**
Versión	1.1 - 08/01/2018 (rev. 13/02/2018)
Descripción	El sistema debe *almacenar la siguiente información sobre cada alerta persistida en la BC:* ▪ *Identificador de la alerta* ▪ *Código hash de la evidencia* ▪ *Evento del sensor que origina la alerta* ▪ *Marca temporal (timestamp)* ▪ *Link de la evidencia* ▪ *Participante que registra la alerta* ▪ *Sensor que origina la alerta*

Tabla 3.138: RI.3 - Alerta en BC

RI.4	
Nombre	**Usuario en BC**
Versión	1.1 - 08/01/2018 (rev. 13/02/2018)
Descripción	El sistema debe *almacenar la siguiente información sobre cada alerta persistida en la BC:* ▪ *Nombre de usuario* ▪ *Dirección email* ▪ *Primer apellido* ▪ *Segundo apellido*

Tabla 3.139: RI.4 - Usuario en BC

RI.5	
Nombre	**Publicar alerta en BC**
Versión	1.1 - 08/01/2018 (rev. 13/02/2018)
Descripción	El sistema debe *almacenar la siguiente información sobre cada transacción registrada en la BC:* ▪ *Identificador de la alerta* ▪ *Código hash de la evidencia* ▪ *Evento del sensor que origina la alerta* ▪ *Marca temporal (timestamp)* ▪ *Link de la evidencia* ▪ *Participante que registra la alerta* ▪ *Sensor que origina la alerta*

Tabla 3.140: RI.5 - Publicar alerta en BC

RI.6	
Nombre	**Tipos de usuario del sistema web**
Versión	1.0 - 08/01/2018
Descripción	El sistema web debe *categorizar y distinguir dos tipos de usuario:* ▪ *Usuario administrador* ▪ *Usuario normal*

Tabla 3.141: RI.6 - Tipos de usuario del sistema web

RI.7	
Nombre	**Usuario administrador**
Versión	1.0 - 08/01/2018
Descripción	El sistema web debe *almacenar la siguiente información sobre cada usuario administrador:* ▪ *Identificador de usuario administrador* ▪ *Contraseña* ▪ *Contraseña expirada* ▪ *Cuenta activa* ▪ *Cuenta bloqueada* ▪ *Cuenta expirada* ▪ *Email* ▪ *Fecha de creación* ▪ *Imagen de perfil* ▪ *Nombre de usuario* ▪ *Tipo de la imagen de perfil*

Tabla 3.142: RI.7 - Usuario administrador

RI.8	
Nombre	**Usuario normal**
Versión	1.0 - 08/01/2018
Descripción	El sistema web debe *almacenar la siguiente información sobre cada usuario normal:* ▪ *Identificador de usuario normal* ▪ *Apellidos* ▪ *Contraseña* ▪ *Contraseña expirada* ▪ *Cuenta activa* ▪ *Cuenta bloqueada* ▪ *Cuenta expirada* ▪ *Email* ▪ *Fecha de creación* ▪ *Imagen de perfil* ▪ *Nombre* ▪ *Nombre de usuario* ▪ *Tipo de la imagen de perfil*

Tabla 3.143: RI.8 - Usuario normal

RI.9	
Nombre	**Rol**
Versión	1.0 - 08/01/2018
Descripción	El sistema web debe *almacenar la siguiente información sobre cada rol:* ▪ *Identificador de rol* ▪ *Nombre de rol*

Tabla 3.144: RI.9 - Rol

RI.10	
Nombre	***Token***
Versión	1.1 - 08/01/2018 (rev. 13/02/2018)
Descripción	El sistema web debe *almacenar la siguiente información sobre cada token de seguridad:* ▪ *Identificador del token* ▪ *Estado* ▪ *Fecha de creación* ▪ *Tipo* ▪ *Token*

Tabla 3.145: RI.10 - Token

3.2. Casos de uso

3.2.1. Actores

Los actores son un conjunto coherente de roles que los usuarios involucrados en el sistema desempeñan al interactuar con él. Un rol puede ser desempeñado por personas físicas, dispositivos *hardware* u otros sistemas. Las Figuras 3.1, 3.2 y 3.3 detallan los actores identificados para el presente sistema:

Figura 3.1: Actor - Usuario administrador.

ACT-0001	Usuario administrador
Versión	1.0 (09/01/2018)
Autores	• Jesús Iglesias García
Fuentes	• Jesús Iglesias García
Descripción	Este actor representa *al usuario del sistema web que accede al panel de control para la gestión de usuarios y la visualización en tiempo real de toda la información de monitorización del entorno. Este usuario también puede ejecutar el resto de componentes del proyecto.*
Comentarios	Requisito: RI.7

Figura 3.2: Actor - Usuario normal.

ACT-0002	Usuario normal
Versión	1.0 (09/01/2018)
Autores	• Jesús Iglesias García
Fuentes	• Jesús Iglesias García
Descripción	Este actor representa *al usuario del sistema web que consume la información en tiempo real de monitorización del entorno de la que sea poseedor. Este usuario también puede ejecutar el resto de componentes del proyecto.*
Comentarios	Requisito: RI.8

Figura 3.3: Actor - Usuario no registrado.

ACT-0003	Usuario sin registrar
Versión	1.0 (09/01/2018)
Autores	• Jesús Iglesias García
Fuentes	• Jesús Iglesias García
Descripción	Este actor representa *al usuario que no posee acceso al sistema web. Este usuario también puede ejecutar el resto de componentes del proyecto.*
Comentarios	Ninguno

3.2.2. Lista de casos de uso

El siguiente listado muestra los casos de uso identificados:

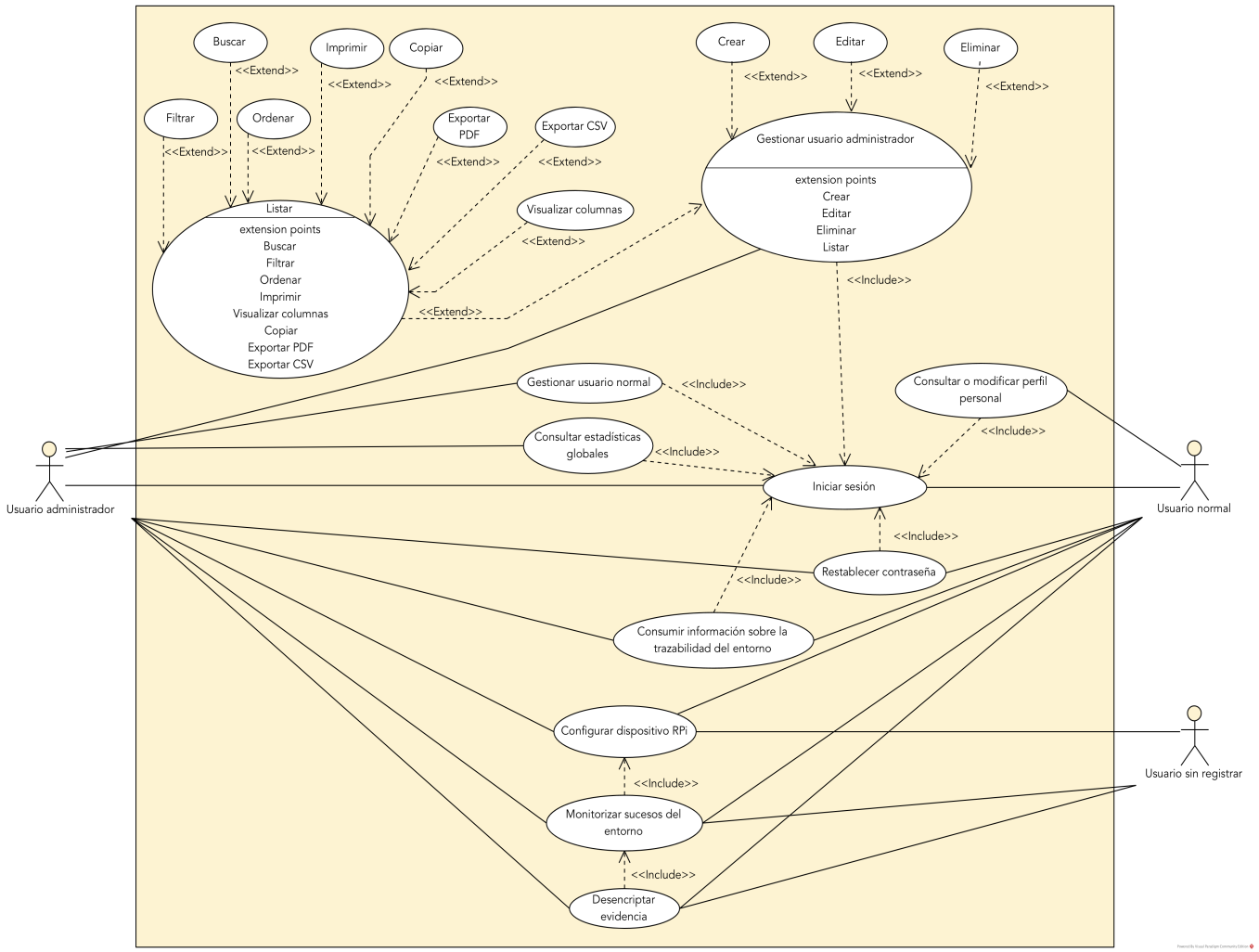
- **UC-0001:** Configurar dispositivo RPi
- **UC-0002:** Monitorizar sucesos del entorno
- **UC-0003:** Desencriptar evidencia
- **UC-0004:** Iniciar sesión
- **UC-0005:** Restablecer contraseña
- **UC-0006* :** Gestionar usuario administrador
- **UC-0007* :** Gestionar usuario normal
- **UC-0008:** Consultar estadísticas globales
- **UC-0009:** Consultar o modificar perfil personal
- **UC-0010:** Consumir información sobre la trazabilidad del entorno

* **¡Importante! -** Como se ha indicado anteriormente, con fines de reducir aquellas partes repetitivas solamente se va a detallar de forma completa el caso de uso UC-0006: Gestionar usuario administrador. El caso de uso UC-0007: Gestionar usuario normal será similar con la salvedad de su adaptación. La gestión de cada entidad engloba las actividades principales de listado, creación, edición, eliminación e importación. A su vez, la actividad listado engloba una serie de subactividades: búsqueda, ordenación, filtrado, elección de columnas (campos de una entidad) a visualizar, restablecimiento de la visualización de las columnas principales, exportación a PDF y CSV, impresión y copiado.

3.2.3. Diagrama de casos de uso

El diagrama de casos de uso (Figura 3.4) tiene como finalidad documentar el comportamiento de un sistema completo desde el punto de vista del usuario.

Figura 3.4: Diagrama de casos de uso.



3.2.4. Descripción de casos de uso

A continuación se detalla el comportamiento de forma superficial de cada caso de uso identificado, omitiendo en algunos casos todas las acciones realizadas y englobándolas en una acción conjunta con el fin de no alargar demasiado la descripción.

UC-0001	**Configurar dispositivo RPi**
Versión	1.0 - 12/01/2018 (rev. 15/02/2018)
Autores	Jesús Iglesias García
Usuario	Actor Usuario administrador (ACT-0001), actor Usuario normal (ACT-0002) o actor Usuario sin registrar (ACT-0003)

Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando *cualquiera de los actores identificados quiera realizar la configuración inicial del dispositivo RPi*	
Pre-condición	El actor se encuentra frente al prototipo *hardware* correctamente conectado y funcional	
Secuencia	**Paso**	**Acción**
	1	El caso de uso comienza cuando cualquier de los actores ejecuta este componente
	2	El componente realiza una serie de comprobaciones: fichero de utilidades, usuario superprivilegiado, plataforma GNU/Linux, dispositivo RPi, conexión de red y concurrencia
	3	El componente comprueba las opciones introducidas al ejecutarlo
	4	El componente ejecuta la instalación de cada paquete si no se encuentra ya instalado y si existe en el repositorio
	5	El componente ejecuta la instalación de cada librería Python si no se encuentra ya instalada y si existe en el repositorio
	6	El componente comprueba que el comando para habilitar interfaces está instalado. En caso afirmativo, comprueba si cada interfaz se encuentra deshabilitada en cuyo caso efectúa la activación de cada una
	7	El componente pregunta al usuario si desea reiniciar el dispositivo
	8	El actor confirma la acción de reiniciar
	9	El componente comprueba que el comando para reiniciar se encuentra instalado y en caso afirmativo reinicia el dispositivo tras 5 segundos de espera
Flujo alternativo	**Paso**	**Acción**
	3.1	El componente comprueba las opciones introducidas detectando que el usuario desea mostrar la ayuda. En dicho caso, muestra la sección informativa, y a continuación el caso de uso finaliza
	3.2	El componente comprueba las opciones introducidas detectando que el usuario desea mostrar más información sobre las acciones ejecutadas. En dicho caso, activa el modo *verbose*, y a continuación el caso de uso continúa en el paso 4
	3.3	El componente comprueba las opciones introducidas detectando que el usuario desea ejecutar solamente la instalación de dependencias. En dicho caso, el caso de uso ejecuta solo los pasos 4, 5 y 6

	3.4	El componente comprueba las opciones introducidas detectando que el usuario desea ejecutar solamente la activación de interfaces. En dicho caso, el caso de uso continúa en el paso 7
	4.1	El componente verifica si el paquete instalado se encuentra desactualizado. En caso afirmativo, es actualizado
	4.2	El componente verifica que cada paquete se encuentra ya instalado y actualizado por lo que no se realiza ninguna acción y el caso de uso continúa en el paso 5
	5.1	El componente verifica si la librería instalada se encuentra desactualizada. En caso afirmativo, es actualizada
	5.2	El componente verifica que cada librería se encuentra ya instalada y actualizada por lo que no se realiza ninguna acción y el caso de uso continúa en el paso 6
	6.1	Si el componente comprueba que ambas interfaces ya se encuentran habilitadas, no se realiza la activación y el caso de uso continúa en el paso 8
	8.1	El actor indica que no desea reiniciar el dispositivo, y a continuación el caso de uso finaliza
Post-condición	El actor ha configurado correctamente el dispositivo y puede entonces ejecutar el componente de monitorización de sucesos del entorno	
Excepciones	**Paso**	**Acción**
	2	Si alguna de las condiciones iniciales no se cumple, el componente notifica al usuario por el terminal y a continuación, el caso de uso finaliza
	3	Si el número de opciones introducido es erróneo o alguna opción no es reconocida, el componente notifica al usuario por el terminal, y a continuación el caso de uso finaliza
	4	Si el paquete no instalado no existe en el repositorio o se produce un error durante la instalación, se notifica al usuario por el terminal y el caso de uso continúa en el paso 4 con el siguiente paquete
	4.1	Si el componente obtiene algún error durante la actualización de un paquete, se notifica al usuario por el terminal y el caso de uso continúa en el paso 4 con el siguiente paquete
	5	Si la librería no instalada no existe en el repositorio o se produce un error durante la instalación, se notifica al usuario por el terminal y el caso de uso continúa en el paso 5 con la siguiente librería
	5.1	Si el componente obtiene algún error durante la actualización de una librería, se notifica al usuario por el terminal y el caso de uso continúa en el paso 5 con la siguiente librería

	6	Si el componente detecta que el comando para habilitar interfaces no se encuentra instalado o se produce un error durante la activación de alguna interfaz, se muestra una notificación de error y el caso de uso finaliza
	9	Si el componente detecta que el comando para reiniciar el dispositivo no se encuentra instalado, se muestra una notificación de error y el caso de uso finaliza
Comentarios	Requisitos: RF.1, RF.2, RF.3, RF.4, RF.5, RF.65, RF.66, RF.67, RF.73	

Tabla 3.146: Descripción del caso de uso - Configurar dispositivo RPi

UC-0002	**Monitorizar sucesos del entorno**	
Versión	1.0 - 12/01/2018 (rev. 15/02/2018)	
Autores	Jesús Iglesias García	
Usuario	Actor Usuario administrador (ACT-0001), actor Usuario normal (ACT-0002) o actor Usuario sin registrar (ACT-0003)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando *cualquiera de los actores identificados quiera monitorizar los sucesos que se producen en el entorno*	
Pre-condición	El actor posee el prototipo *hardware* correctamente conectado y configurado	
Secuencia	**Paso**	**Acción**
	1	El caso de uso comienza cuando cualquiera de los actores ejecuta este componente
	2	El componente realiza una serie de comprobaciones iniciales: usuario superprivilegiado, plataforma GNU/Linux, dispositivo RPi, conexión de red y concurrencia
	3	El componente inicializa el *logger*
	4	El componente comprueba las opciones introducidas al ejecutarlo
	5	El componente solicita las credenciales y datos de configuración de las distintas utilidades y servicios
	6	El usuario introduce la información
	7	El componente inicializa cada utilidad y servicio
	8	El componente monitoriza los sucesos del entorno mediante la obtención de una medición, mostrando la información por el terminal y los LCDs
	9	El componente comprueba si la medición actual representa un suceso anómalo. En caso afirmativo, activa el protocolo de alerta

	10	El componente efectúa las acciones del protocolo de alerta: capturar evidencia (vídeo), calcular código *hash*, encriptar y firmar la evidencia, almacenar la evidencia en la nube, almacenar medición en la BBDD, registrar la alerta en la BC y notificar por *email* en caso de que dicha opción esté activa
	11	El caso de uso vuelve al paso 8
Flujo alternativo	**Paso**	**Acción**
	4.1	El componente comprueba las opciones introducidas detectando que el usuario desea mostrar la ayuda. En dicho caso, muestra la sección informativa, y a continuación el caso de uso finaliza
	9.1	El componente comprueba si la medición actual representa un suceso anómalo. En caso negativo, no activa ningún protocolo
	9.1.1	El componente efectúa la acción de una medición normal: almacenar medición en la BBDD
	9.1.2	El caso de uso vuelve al paso 8
	9.2	El componente comprueba si la medición actual es errónea. En caso afirmativo, no muestra ningún valor por el terminal ni los LCDs y a continuación, el caso de uso vuelve al paso 8
Post-condición	El actor ha monitorizado el entorno IoT registrando tanto los sucesos normales como aquellos considerados anómalos	
Excepciones	**Paso**	**Acción**
	2	Si alguna de las condiciones iniciales no se cumple, el componente notifica al usuario por el terminal y a continuación, el caso de uso finaliza
	3	Si el componente obtiene algún error durante la creación del directorio donde se almacenará el fichero de *log*, se notifica al usuario por el terminal y a continuación, el caso de uso finaliza
	4	Si el número de opciones introducido es erróneo, alguna opción presenta un valor incorrecto o no es reconocida, el componente notifica al usuario por el terminal y a continuación, el caso de uso finaliza
	7	Si el componente obtiene algún error durante la inicialización de alguna utilidad o servicio, se notifica al usuario por el terminal y a continuación, el caso de uso finaliza
	9.1.1	Si el componente obtiene algún error durante la ejecución de alguna acción tras una medición normal, se notifica al usuario por el terminal, se envía un *email* del error en caso de estar activa dicha opción y a continuación, el caso de uso finaliza

	10	Si el componente obtiene algún error durante la ejecución de alguna acción del protocolo de alerta, se notifica al usuario por el terminal, se envía un *email* del error en caso de estar activa dicha opción y a continuación, el caso de uso finaliza
Comentarios	Requisitos: RF.6, RF.7, RF.8, RF.9, RF.10, RF.11, RF.12, RF.13, RF.14, RF.15, RF.16, RF.17, RF.18, RF.19, RF.20, RF.21, RF.22, RF.23, RF.24, RF.25, RF.26, RF.27, RF.28, RF.29, RF.30, RF.31, RF.32, RF.33, RF.34, RF.35, RF.36, RF.37, RF.38, RF.39, RF.40, RF.41, RF.42, RF.43, RF.44, RF.45, RF.46, RF.47, RF.48, RF.49, RF.50, RF.51, RF.57, RF.58, RF.64, RF.65, RF.66, RF.68, RF.69, RF.70, RF.71, RF.72, RF.73	

Tabla 3.147: Descripción del caso de uso - Monitorizar sucesos del entorno

UC-0003	**Desencriptar evidencia**	
Versión	1.0 - 12/01/2018 (rev. 15/02/2018)	
Autores	Jesús Iglesias García	
Usuario	Actor Usuario administrador (ACT-0001), actor Usuario normal (ACT-0002) o actor Usuario sin registrar (ACT-0003)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando *cualquiera de los actores identificados quiera desencriptar una evidencia previamente encriptada con GPG, verificar la firma y la integridad de su contenido*	
Pre-condición	El actor posee la evidencia, acceso al par de claves usado durante la encriptación y firmado y el código *hash* del contenido de la evidencia	
Secuencia	**Paso**	**Acción**
	1	El caso de uso comienza cuando cualquiera de los actores ejecuta este componente
	2	El componente comprueba que la ejecución se efectuó con un usuario superprivilegiado
	3	El componente comprueba las opciones introducidas al ejecutarlo
	4	El componente descarga de la nube la evidencia encriptada y firmada en caso de introducir la opción correspondiente
	5	El componente comprueba la existencia en el sistema local de los ficheros (evidencia, par de claves) y directorios indicados en las opciones
	6	El componente comprueba que la evidencia posea el formato correcto
	7	El componente importa el par de claves desde el fichero indicado
	8	El componente solicita la contraseña de protección de la clave privada
	9	El actor introduce la credencial
	10	El componente comprueba que el dato introducido no sea vacío

	11	El componente desencripta la evidencia
	12	El componente verifica la firma
	13	El componente calcula el código *hash* de la evidencia recién desencriptada y lo compara con el valor *hash* introducido
Flujo alternativo	**Paso**	**Acción**
	3.1	El componente comprueba las opciones introducidas detectando que el usuario desea mostrar la ayuda. En dicho caso, muestra la sección informativa, y a continuación el caso de uso finaliza
	7.1	El componente comprueba que la huella digital o *fingerprint* introducida se relacione con el de alguna clave existente en el almacén de claves
	12.1	El componente verifica que la evidencia no fue firmada
Post-condición	El actor ha desencriptado la evidencia y ha comprobado la firma y la integridad del contenido	
Excepciones	**Paso**	**Acción**
	2	El componente detecta que la ejecución fue con un usuario normal, por lo que notifica al usuario del error, y a continuación el caso de uso finaliza
	3	Si el número de opciones introducido es erróneo, no se ha introducido alguna opción obligatoria, alguna opción presenta un valor incorrecto o no es reconocida, el componente notifica al usuario por el terminal y a continuación, el caso de uso finaliza
	5	Si el componente verifica que algún fichero o directorio no existe o no es del tipo correcto, notifica al usuario por el terminal, y a continuación el caso de uso finaliza
	6	Si el componente verifica que la evidencia posee un formato distinto al permitido, notifica al usuario por el terminal, y a continuación el caso de uso finaliza
	7	Si el componente verifica que el fichero no contiene el par de claves, notifica al usuario por el terminal, y a continuación el caso de uso finaliza
	7.1	Si el componente verifica que el almacén de claves no contiene ninguna clave o el *fingerprint* no se relaciona con el de alguna clave existente en el almacén, notifica al usuario por el terminal, y a continuación el caso de uso finaliza

	10	Si el componente verifica que la credencial introducida son espacios en blanco, notifica al usuario por el terminal siempre y cuando haya gastado el número de intentos (3), y a continuación el caso de uso finaliza
	11	Si el componente obtiene algún error durante la desencriptación, notifica al usuario por el terminal, y a continuación el caso de uso finaliza
	13	Si el componente obtiene algún error durante el cálculo del valor *hash*, notifica al usuario por el terminal, y a continuación el caso de uso finaliza
Comentarios	Requisitos: RF.52, RF.53, RF.54, RF.55, RF.56, RF.57, RF.58, RF.59, RF.60, RF.61, RF.62, RF.63, RF.64, RF.65, RF.66, RF.70, RF.71, RF.73	

Tabla 3.148: Descripción del caso de uso - Desencriptar evidencia

UC-0004	**Iniciar sesión**	
Versión	1.0 - 12/01/2018 (rev. 16/02/2018)	
Autores	Jesús Iglesias García	
Usuario	Actor Usuario administrador (ACT-0001) o actor Usuario normal (ACT-0002)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando *un actor Usuario administrador (ACT-0001) o actor Usuario normal (ACT-0002) quiera iniciar sesión en el sistema web*	
Pre-condición	El actor Usuario administrador (ACT-0001) o actor Usuario normal (ACT-0002) se encuentra en la página de inicio del sistema web	
Secuencia	**Paso**	**Acción**
	1	El caso de uso comienza cuando un actor Usuario administrador (ACT-0001) o actor Usuario normal (ACT-0002) introduce sus credenciales para la autenticación
	2	El sistema web comprueba que los datos introducidos sean correctos. En caso afirmativo, crea la sesión de usuario y le redirige a la página de usuario dentro del sistema, si éste posee rol usuario normal
Flujo alternativo	**Paso**	**Acción**
	1.1	El sistema web comprueba si existe la *cookie* referente a la funcionalidad de *Recordarme*. En caso afirmativo, el sistema web inicia sesión automáticamente al usuario y le redirige a la página de usuario dentro del sistema o al panel de control según su rol. A continuación, el caso de uso finaliza

	2.1	El sistema web comprueba que los datos introducidos sean correctos. En caso afirmativo, crea la sesión de usuario y le redirige al panel de control, si éste posee rol usuario administrador
Post-condición	El actor Usuario administrador (ACT-0001) o actor Usuario normal (ACT-0002) ha iniciado sesión en el sistema web y se encuentra dentro de él ya sea en el panel de control o en la página de usuario según el rol de éste	
Excepciones	**Paso**	**Acción**
	1.1	Si la *cookie* referente a la funcionalidad de *Recordarme* no existe, el sistema web redirige a la página de inicio, y a continuación el caso de uso finaliza
	2, 2.1	Si el sistema web obtiene algún error al comprobar los datos introducidos (datos erróneos, usuario no registrado, cuenta bloqueada, cuenta sin activar, cuenta o contraseña expirada), muestra un mensaje de error, y a continuación el caso de uso finaliza.
Comentarios	Requisitos: RF.74, RF.75, RF.76, RF.77	

Tabla 3.149: Descripción del caso de uso - Iniciar sesión

UC-0005	**Restablecer contraseña**	
Versión	1.0 - 13/01/2018 (rev. 16/02/2018)	
Autores	Jesús Iglesias García	
Usuario	Actor Usuario administrador (ACT-0001) o actor Usuario normal (ACT-0002)	
Descripción	El sistema web deberá comportarse tal como se describe en el siguiente caso de uso cuando *un actor Usuario administrador (ACT-0001) o actor Usuario normal (ACT-0002) quiera restablecer su contraseña*	
Pre-condición	El actor Usuario administrador (ACT-0001) o actor Usuario normal (ACT-0002) se encuentra en la página de inicio del sistema web	
Secuencia	**Paso**	**Acción**
	1	El caso de uso comienza cuando un actor Usuario administrador (ACT-0001) o actor Usuario normal (ACT-0002) selecciona la opción para restablecer la contraseña
	2	El sistema web muestra el formulario a rellenar
	3	El actor Usuario administrador (ACT-0001) o actor Usuario normal (ACT-0002) introduce su *email* asociado a su cuenta existente

	4	El sistema web comprueba que el *email* introducido es correcto. En caso afirmativo, envía al usuario un correo electrónico de restablecimiento de la contraseña y le avisa mediante un mensaje
	5	El actor Usuario administrador (ACT-0001) o actor Usuario normal (ACT-0002) restablece su contraseña a través del correo de restablecimiento
	6	El sistema comprueba que la nueva contraseña sea correcta. En caso afirmativo, asigna la nueva contraseña al usuario y muestra la página de inicio del sistema
Post-condición	El actor Usuario administrador (ACT-0001) o actor Usuario normal (ACT-0002) ha modificado su contraseña	
Excepciones	**Paso**	**Acción**
	3	Si el actor Usuario administrador (ACT-0001) o actor Usuario normal (ACT-0002) cancela la introducción del *email* en el formulario, el sistema web no realiza ninguna acción, y a continuación el caso de uso finaliza
	4	Si el sistema web obtiene algún error al comprobar el *email* introducido (dirección de correo inexistente), muestra un mensaje, y a continuación el caso de uso vuelve al paso 2
	4	Si el sistema web obtiene algún error al enviar el *email*, muestra un mensaje de error y el caso de uso finaliza
	6	Si el sistema web obtiene algún error al comprobar el *token* de seguridad (expirado, ya usado o de distinto tipo) de restablecimiento de contraseña, muestra un mensaje de error y el caso de uso finaliza
	6	Si el sistema web obtiene algún error al comprobar la nueva contraseña (no cumple con un patrón), muestra un mensaje de error, y a continuación el caso de uso vuelve al paso 5
Comentarios	Requisitos: RF.78	

Tabla 3.150: Descripción del caso de uso - Restablecer contraseña

UC-0006	**Gestionar usuario administrador ***
Versión	1.0 - 13/01/2018 (rev. 16/02/2018)
Autores	Jesús Iglesias García
Usuario	Actor Usuario administrador (ACT-0001)
Descripción	El sistema web deberá comportarse tal como se describe en el siguiente caso de uso cuando *un actor Usuario administrador (ACT-0001)* quiera gestionar un usuario administrador

Pre-condición	El actor Usuario administrador (ACT-0001) debe estar autenticado en el sistema web	
Secuencia	**Paso**	**Acción**
	1	El caso de uso comienza cuando el actor Usuario administrador (ACT-0001) notifica desde el panel de control que desea gestionar un usuario administrador
	2	El sistema web muestra un listado ordenado y paginado o filtrado de los usuarios administradores disponibles en el sistema
	3	El actor Usuario administrador (ACT-0001) selecciona el apartado para crear un nuevo usuario administrador
	4	El sistema web muestra el formulario con los campos a rellenar del nuevo usuario
	5	El actor Usuario administrador (ACT-0001) completa los datos del formulario
	6	El sistema web comprueba que los datos introducidos sean correctos. En caso afirmativo, almacena el nuevo usuario con rol usuario administrador y avisa al actor Usuario administrador (ACT-0001) mediante un mensaje. A continuación, el caso de uso vuelve al paso 2
Flujo alternativo	**Paso**	**Acción**
	3.1.1	El actor Usuario administrador (ACT-0001) selecciona un usuario administrador existente para su edición
	3.1.2	El sistema web muestra el formulario con los datos del usuario administrador seleccionado
	3.1.3	El actor Usuario administrador (ACT-0001) edita los campos deseados
	3.1.4	El sistema web comprueba que los datos editados sean correctos. En caso afirmativo, almacena el usuario modificado y avisa al actor Usuario administrador (ACT-0001) mediante un mensaje. A continuación, el caso de uso vuelve al paso 2
	3.2.1	El actor Usuario administrador (ACT-0001) selecciona un usuario administrador existente para eliminarle
	3.2.2	El sistema web muestra un *pop-up* de confirmación
	3.2.3	El actor Usuario administrador (ACT-0001) confirma la eliminación
	3.2.4	El sistema web elimina el usuario administrador seleccionado y avisa al actor Usuario administrador (ACT-0001) mediante un mensaje. A continuación, el caso de uso vuelve al paso 2

	3.3.1	El actor Usuario administrador (ACT-0001) introduce una cadena de caracteres para realizar una búsqueda entre todos los usuarios administradores existentes. A continuación, el caso de uso vuelve al paso 2
	3.4.1	El actor Usuario administrador (ACT-0001) selecciona un criterio de ordenación. A continuación, el caso de uso vuelve al paso 2
	3.5.1	El actor Usuario administrador (ACT-0001) filtra el número de usuarios administradores a mostrar. A continuación, el caso de uso vuelve al paso 2
	3.6.1	El actor Usuario administrador (ACT-0001) selecciona las columnas (campos de la entidad) que desea mostrar. A continuación, el caso de uso vuelve al paso 2
	3.7.1	El actor Usuario administrador (ACT-0001) selecciona la opción de exportar datos en PDF o CSV
	3.7.2	El sistema web genera el fichero con la información. A continuación, el caso de uso vuelve al paso 2
	3.8.1	El actor Usuario administrador (ACT-0001) selecciona la opción de imprimir
	3.8.2	El sistema web genera la funcionalidad para permitir al usuario imprimir la información de los usuarios administradores. A continuación, el caso de uso vuelve al paso 2
	3.9.1	El actor Usuario administrador (ACT-0001) selecciona la opción de copiar
	3.9.2	El sistema web copia las información de los usuarios administradores al portapapeles. A continuación, el caso de uso vuelve al paso 2
Post-condición	El actor Usuario administrador (ACT-0001) ha administrado un usuario administrador a través de la posibilidad de ejecutar diferentes tareas	
Excepciones	**Paso**	**Acción**
	2	Si el sistema web obtiene algún error al cargar los usuarios administradores, muestra un mensaje de error, y a continuación el caso de uso finaliza.
	3.1.2	Si el sistema web obtiene algún error al cargar los datos del usuario administrador seleccionado, muestra un mensaje de error, y a continuación el caso de uso finaliza
	5, 3.1.3	Si el actor Usuario administrador (ACT-0001) cancela la introducción o edición de datos, el sistema web no almacena ningún dato, y a continuación el caso de uso vuelve al paso 2

	6, 3.1.4	Si el sistema web obtiene algún error al comprobar los datos introducidos o editados (datos erróneos, nombre de usuario o *email* ya existentes, etc.), muestra un mensaje de error, y a continuación, el caso de uso vuelve al paso 4 o 3.1.2, respectivamente
	3.2.3	El actor Usuario administrador (ACT-0001) cancela la confirmación de la eliminación. El sistema web no realiza ninguna acción, y a continuación el caso de uso vuelve al paso 2
	3.2.4	Si el sistema web obtiene algún error al borrar el usuario administrador, muestra un mensaje de error, y a continuación el caso de uso vuelve al paso 2
Comentarios	Requisitos: RF.79, RF.80, RF.81, RF.82, RF.83, RF.84, RF.85, RF.86, RF.87, RF.88, RF.89, RF.90, RF.91, RF.92, RF.93	

Tabla 3.151: Descripción del caso de uso - Gestionar usuario administrador

UC-0008	**Consultar estadísticas globales**	
Versión	1.0 - 15/01/2018	
Autores	Jesús Iglesias García	
Usuario	Actor Usuario administrador (ACT-0001)	
Descripción	El sistema web deberá comportarse tal como se describe en el siguiente caso de uso cuando *un actor Usuario administrador (ACT-0001) quiera consultar las estadísticas de toda la información monitorizada*	
Pre-condición	El actor Usuario administrador (ACT-0001) debe estar autenticado en el sistema web	
Secuencia	**Paso**	**Acción**
	1	El caso de uso comienza cuando el actor Usuario administrador (ACT-0001) notifica desde el panel de control que desea ver las estadísticas actuales de la información completa monitorizada
	2	El sistema web carga los diferentes gráficos y datos de estadísticas
Post-condición	El actor Usuario administrador (ACT-0001) puede analizar las diferentes estadísticas	
Excepciones	**Paso**	**Acción**
	2	Si el sistema web obtiene algún error al cargar las estadísticas, muestra un mensaje de error, y a continuación el caso de uso finaliza
Comentarios	Requisitos: RF.94	

Tabla 3.152: Descripción del caso de uso - Consultar estadísticas globales

UC-0009	**Consultar o modificar perfil personal**	
Versión	1.0 - 15/01/2018	
Autores	Jesús Iglesias García	
Usuario	Actor Usuario normal (ACT-0002)	
Descripción	El sistema web deberá comportarse tal como se describe en el siguiente caso de uso cuando *un actor Usuario normal (ACT-0002) quiera consultar o modificar su perfil personal*	
Pre-condición	El actor Usuario normal (ACT-0002) debe estar autenticado en el sistema web	
Secuencia	**Paso**	**Acción**
	1	El caso de uso comienza cuando el actor Usuario normal (ACT-0002) notifica que desea acceder a su panel de perfil personal
	2	El sistema web muestra el perfil personal del usuario con los datos de éste
	3	El actor Usuario normal (ACT-0002) edita los campos deseados
	4	El sistema web comprueba que los datos editados sean correctos. En caso afirmativo, almacena el usuario modificado y notifica al actor Usuario normal (ACT-0002) mediante un mensaje
Post-condición	El actor Usuario normal (ACT-0002) ha consultado o modificado su perfil personal	
Excepciones	**Paso**	**Acción**
	2	Si el sistema web obtiene algún error al cargar los datos del usuario, muestra un mensaje de error, y a continuación el caso de uso finaliza
	3	Si el actor Usuario normal (ACT-0002) cancela la edición de datos, el sistema web no almacena ningún dato, y a continuación el caso de uso finaliza
	4	Si el sistema web obtiene algún error al comprobar los datos editados (datos erróneos, nombre de usuario o *email* ya existentes, etc.), muestra un mensaje de error, y a continuación el caso de uso vuelve al paso 2
Comentarios	Requisitos: RF.95, RF.96	

Tabla 3.153: Descripción del caso de uso - Consultar o modificar perfil personal

UC-0010	**Consumir información sobre la trazabilidad del entorno**	
Versión	1.0 - 15/01/2018	
Autores	Jesús Iglesias García	
Usuario	Actor Usuario administrador (ACT-0001) o actor Usuario normal (ACT-0002)	
Descripción	El sistema web deberá comportarse tal como se describe en el siguiente caso de uso cuando *un actor Usuario administrador (ACT-0001) o actor Usuario normal (ACT-0002) quiera consultar la información sobre la trazabilidad del entorno en tiempo real*	
Pre-condición	El actor debe estar autenticado en el sistema web y el prototipo *hardware* estar monitorizando los sucesos del entorno	
Secuencia	**Paso**	**Acción**
	1	El caso de uso comienza cuando el actor Usuario administrador (ACT-0001) o actor Usuario normal (ACT-0002) notifica que desea consultar la información en tiempo real
	2	El sistema web carga la información completa o filtrada en función del rol del usuario
Post-condición	El actor Usuario administrador (ACT-0001) o actor Usuario normal (ACT-0002) puede observar lo que está sucediendo en el entorno IoT monitorizado	
Excepciones	**Paso**	**Acción**
	2	Si el sistema web obtiene algún error al cargar la información, muestra un mensaje de error, y a continuación el caso de uso finaliza
Comentarios	Requisitos: RF.97	

Tabla 3.154: Descripción del caso de uso - Consumir información sobre la trazabilidad del entorno

3.2.5. Escenarios principales

Los escenarios principales son una secuencia de pasos que realiza un actor del sistema o el propio sistema para reflejar el flujo exitoso más habitual de las acciones descritas en los casos de uso.

- Configurar dispositivo RPi:

  Pablo Ruiz, trabajador de una multinacional tecnológica en el área de desarrollo *software* en IoT, ha recibido una propuesta por parte de Alberto -*manager* del área- donde solicita controlar el centro de procesamiento de datos (CPD) ubicado en la planta baja debido a los problemas surgidos últimamente por sobrecalentamiento y manifestados habitualmente a primera hora de la mañana. La propuesta que le hace llegar consiste en trazar la temperatura y humedad del habitáculo que aloja este CPD -compuesto por gran cantidad de

servidores- para conocer en todo momento lo que está sucediendo e intentar dar con la causa de los problemas experimentados. Además, Pablo propone a Alberto controlar también el acceso a esta zona ya que se considera de acceso restringido y sería de gran ayuda saber si se está produciendo algún acceso no autorizado fuera de la jornada laboral.

Pablo conoce Hyot, la prueba de concepto (*Proof of Concept* -PoC-) desarrollada por Jesús Iglesias para la trazabilidad de entornos IoT mediante Hyperledger. Como se adecua perfectamente a la propuesta, decide utilizar este sistema y lo descarga del repositorio de GitHub. Tras ello, lee la documentación y solicita al *manager* la adquisición de los componentes *hardware* necesarios para montar el prototipo.

Tras este primer paso, Pablo se da cuenta que este sistema requiere de una serie de dependencias para un correcto funcionamiento y decide utilizar el componente de configuración inicial del dispositivo que incluye el repositorio en lugar de realizar una instalación manual ya que esto le ahorra tiempo y esfuerzo. Como en un principio desconoce el funcionamiento de este componente decide mostrar la ayuda y observa que hay una opción para mostrar más información sobre las acciones que se irán ejecutando por lo que ejecuta el código activando esta opción. Mientras el código efectúa los diferentes pasos, Pablo observa cómo se instalan las dependencias necesarias y se habilitan las interfaces, notando al final que el sistema le informa de los siguientes pasos y le pide confirmar si desea reiniciar el dispositivo a lo cual accede.

- Monitorizar sucesos del entorno:

  Tras haber realizado la configuración inicial, Pablo prepara la localización dentro del CPD donde va a situar el sistema, decidiendo en ese momento cuáles serán los umbrales más idóneos de los eventos a partir de los cuáles los valores medidos se considerarán anómalos y activando así el protocolo de alerta. Antes de realizar la primera prueba, consulta el manual de configuración y de usuario para conocer los servicios de los que debe disponer como, por ejemplo: cuenta en el servicio de almacenamiento en la nube Dropbox, cuenta Gmail, red de negocio desplegada en la BC, etc.

  En primer lugar, realiza la puesta en marcha y configuración de todos los servicios requeridos y obtiene las direcciones I2C de los LCDs las cuales introducirá como opciones en la ejecución. Posteriormente, decide ejecutar el componente de monitorización inicializando los servicios y estableciendo como umbrales límites: 80 °C de temperatura y 2 metros de distancia. Con estos valores podrá controlar si se produce algún acceso no autorizado durante el horario no laboral al CPD y así intentar resolver ese contratiempo que están sufriendo cada vez más a menudo lo que les provoca una bajada de rendimiento en el ámbito laboral al no funcionar

todos los sistemas, caerse la red continuamente, etc.

- Iniciar sesión:

  Pablo, que tiene acceso al sistema web como usuario administrador, decide que es conveniente iniciar sesión en él para examinar que el prototipo está funcionando correctamente y por lo tanto seguirá funcionando cuando el horario laboral del día finalice y así asegurarse que podrá trazar el entorno durante la tarde y la noche.

  Para ello, se dirige al formulario de iniciar sesión e introduce las credenciales por defecto del sistema. Además, decide no activar la opción de inicio de sesión automático ya que no quiere que otra persona pueda acceder al sistema web en caso de olvido del cierre de sesión. El sistema comprueba las credenciales introducidas y como éstas son correctas permite el acceso a Pablo al panel de control.

- Consumir información sobre la trazabilidad del entorno:

  Pablo en su afán de comprobar que el sistema está funcionando de forma apropiada quiere asegurarse que el entorno está siendo monitorizado para así no llevarse una sorpresa al día siguiente cuando compruebe si se detectó algún suceso anómalo. Por ello, dentro del sistema web se dirige a la sección donde se muestran las mediciones que se van tomando de los eventos sobre el entorno y comprueba que cada 3 segundos una nueva medición normal se va registrando. Como el aspecto relevante es el registro de incidencias, es decir, sucesos anómalos fuerza una lectura de este tipo entrando en el CPD y acercándose a los servidores. De esta forma, el sistema detecta una entrada y un acercamiento que sobrepasa el umbral límite de distancia -establecido en 2 metros- por lo que activa el protocolo de alerta consistente entre otros pasos de la captura de un vídeo por defecto de 10 segundos, su almacenamiento encriptado y firmado en la nube y el registro del valor *hash* -entre otra información- en la BC para proporcionar la prueba irrefutable de que esa evidencia sucedió en ese instante temporal. Posteriormente, Pablo visualiza en el sistema web la sección de alertas y observa como una nueva alerta fue registrada en este mecanismo de persistencia.

- Desencriptar evidencia:

  Tras provocar controladamente la incidencia, Pablo quiere asegurarse que la evidencia -vídeo capturado- refleja su entrada al habitáculo del CPD y por tanto puede confiar en este sistema. En el sistema web, obtiene el enlace para obtener la evidencia encriptada y firmada y el código *hash* de su contenido original, este último dato para verificar la integridad del contenido. Ejecuta el componente de desencriptación indicando el enlace de Dropbox que contiene la evidencia, el código *hash*, el directorio destino y el fichero de claves utilizado durante la encriptación y firmado. Tras su ejecución, comprueba que la evidencia fue descargada del sistema de almacenamiento y fue correctamente desencriptada constatando que

su contenido no fue alterado y verificando el origen de la firma, el par de claves creado por el mismo y asociado a su nombre e *email* empresarial.

Con la evidencia desencriptada, Pablo procede a su visualización y efectivamente la reproducción del vídeo delata a Pablo entrando al CPD por lo que la monitorización del entorno ha sido satisfactoria garantizando además la originalidad del suceso.

- Gestionar usuario administrador:

  Pablo ha podido acceder con un usuario administrador preconfigurado por defecto por lo que cree que no es ideal su existencia. Por ello, decide dar de alta un nuevo administrador dirigiéndose a tal formulario. En él, introduce un nombre de usuario y dirección de correo y comprueba in situ si dichas elecciones se encuentran disponibles en el sistema, es decir, no hay ningún otro usuario registrado con esos datos. Como ambas elecciones están disponibles, continúa rellenando el resto de información y crea el nuevo usuario observando que una notificación es mostrada. Pablo a partir de ese momento ya posee su propio usuario administrador por lo que procede a listar los usuarios administradores del sistema y seleccionar aquel que venía preconfigurado para su eliminación.

- Gestionar usuario normal:

  Pablo comenta a Alberto que ya ha puesto en marcha el sistema y que les permitirá intentar resolver el problema que están experimentando casi a diario. Alberto, muy contento con esta noticia pregunta a Pablo si existe alguna forma donde ver los sucesos registrados del entorno en tiempo real así como las alertas lanzadas. Pablo le comenta la existencia de un sistema web y que con un usuario normal podrá iniciar sesión y ver la monitorización de la información de la que sea poseedor.

  Entonces, Pablo se dirige a la gestión de usuarios normales y crea una nueva cuenta de usuario activa asociada a su *manager* Alberto cuyo nombre de usuario coincide con el usuario de la BC que registra la información. Además, por curiosidad desea mantener una copia externa de estos datos y por ello en el listado selecciona la opción: PDF y posteriormente CSV con la cual obtendrá la descarga de dos ficheros con la información completa. Además, se percata de que existe la opción de imprimir aunque debido a la reciente rotura de la impresora asignada decide no usar esta opción.

- Consultar estadísticas:

  Pablo desde su panel de control observa que existe una sección donde se muestran estadísticas completas sobre el sistema, entre ellas, el número total de usuarios que existen en la plataforma diferenciados por rol, últimas cuentas de usuario normal creadas, número de

mediciones, número de alertas producidas, etc. Visualizando la información se percata que la nueva cuenta de Alberto ya está reflejada en el sistema.

- Restablecer contraseña:

  Alberto, tras una semana sin acceder al sistema web, ya no recuerda su contraseña debido a la variedad de contraseñas que posee de otros servicios. Por ello al dirigirse a la página inicial se percata de que existe una opción para restablecer la contraseña a la cual accede. Allí comprueba que el sistema le solicita su *email* y Alberto encantado de poder restablecer el acceso a su cuenta, introduce su *email* registrado.

  El sistema comprueba que el *email* introducido pertenece a un usuario existente de la plataforma. Como pertenece en este caso al usuario Alberto, el sistema le envía un *email* con una validez de 30 minutos para el restablecimiento de la contraseña. Entonces, Alberto se dirige a su bandeja de entrada del servicio de *email* y abre el correo que acaba de recibir pulsando en el *link* el cual le redirige a una página para introducir la nueva contraseña. Allí, Alberto establece la nueva contraseña cumpliendo los patrones requeridos. Una vez introducida la nueva contraseña, el sistema lo almacena y Alberto ya tiene de nuevo acceso a la plataforma para consultar la información del entorno trazado.

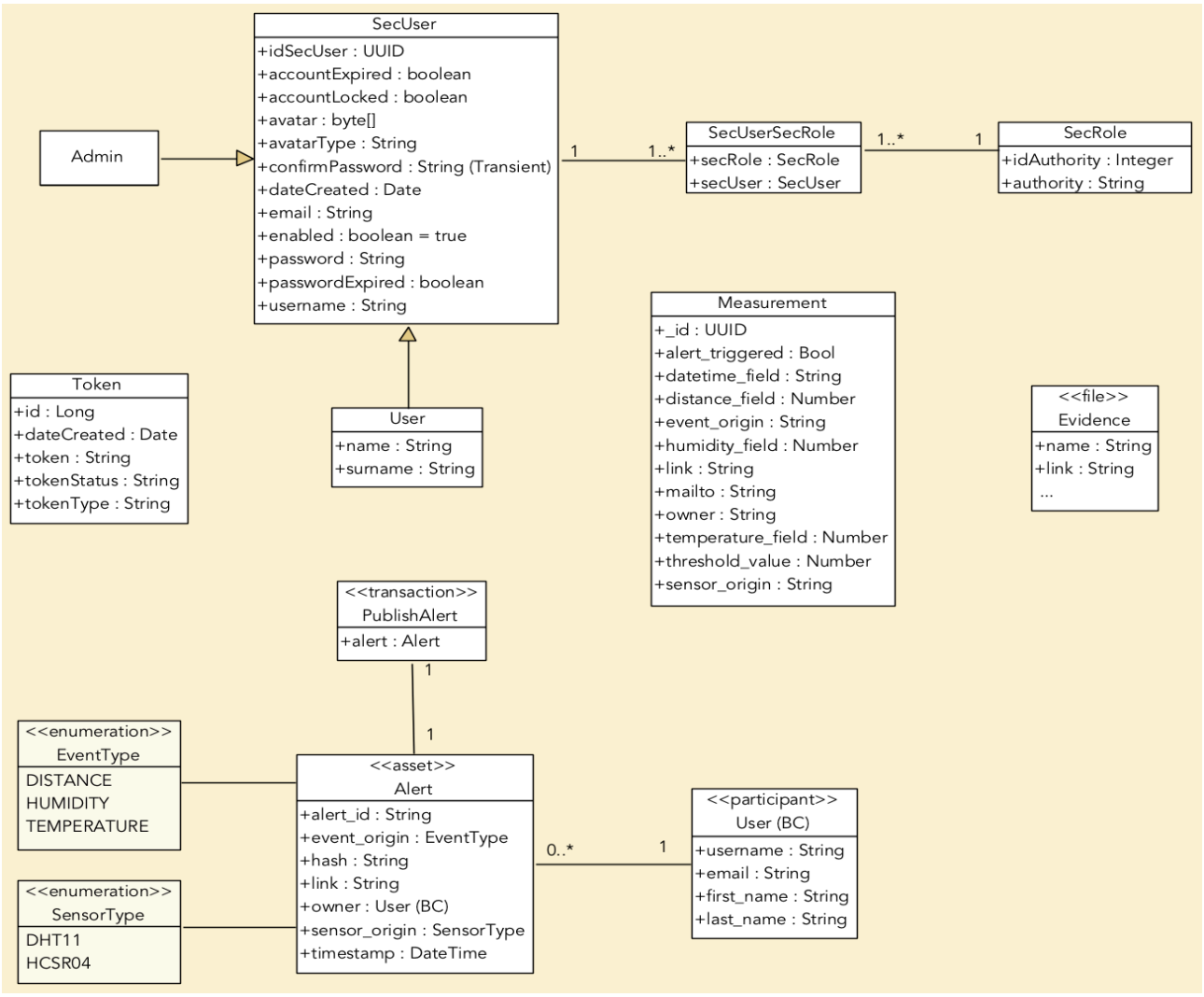
- Consultar o modificar perfil personal:

  Debido a la política restrictiva de modificación de contraseña cada 6 meses impuesta por la empresa, Pablo tuvo que cambiar su contraseña de acceso a los distintos servicios internos. Como cree que es conveniente poseer la misma contraseña en el sistema web para evitar que se le olvide, una vez iniciado sesión se dirige a su perfil personal y allí observa que hay un apartado para modificar la contraseña. Entonces Pablo siguiendo las indicaciones de información sobre el patrón que debe seguir su nueva contraseña, elige una que es segura y entonces rellena el formulario, indicando la antigua y la nueva contraseña. El sistema confirma este cambio y a partir de ahora Pablo puede acceder con la nueva contraseña establecida.

3.3. Modelo de dominio

El modelo de dominio representa visualmente las diferentes clases conceptuales del mundo real en un dominio de interés, describiendo las entidades, atributos y relaciones. La Figura 3.5 muestra este diagrama.

Figura 3.5: Modelo de dominio.



3.4. Diagramas de secuencia de análisis

En esta sección se muestran los diagramas de secuencia de análisis los cuales representan las interacciones de las entidades del sistema con los casos de uso, ambos detallados anteriormente.

- Diagrama de secuencia de análisis: Configurar dispositivo RPi (Figura 3.6 - Parte 1 y Figura 3.7 - Parte 2).

Figura 3.6: Diagrama de secuencia de análisis: Configurar dispositivo RPi - Parte 1.

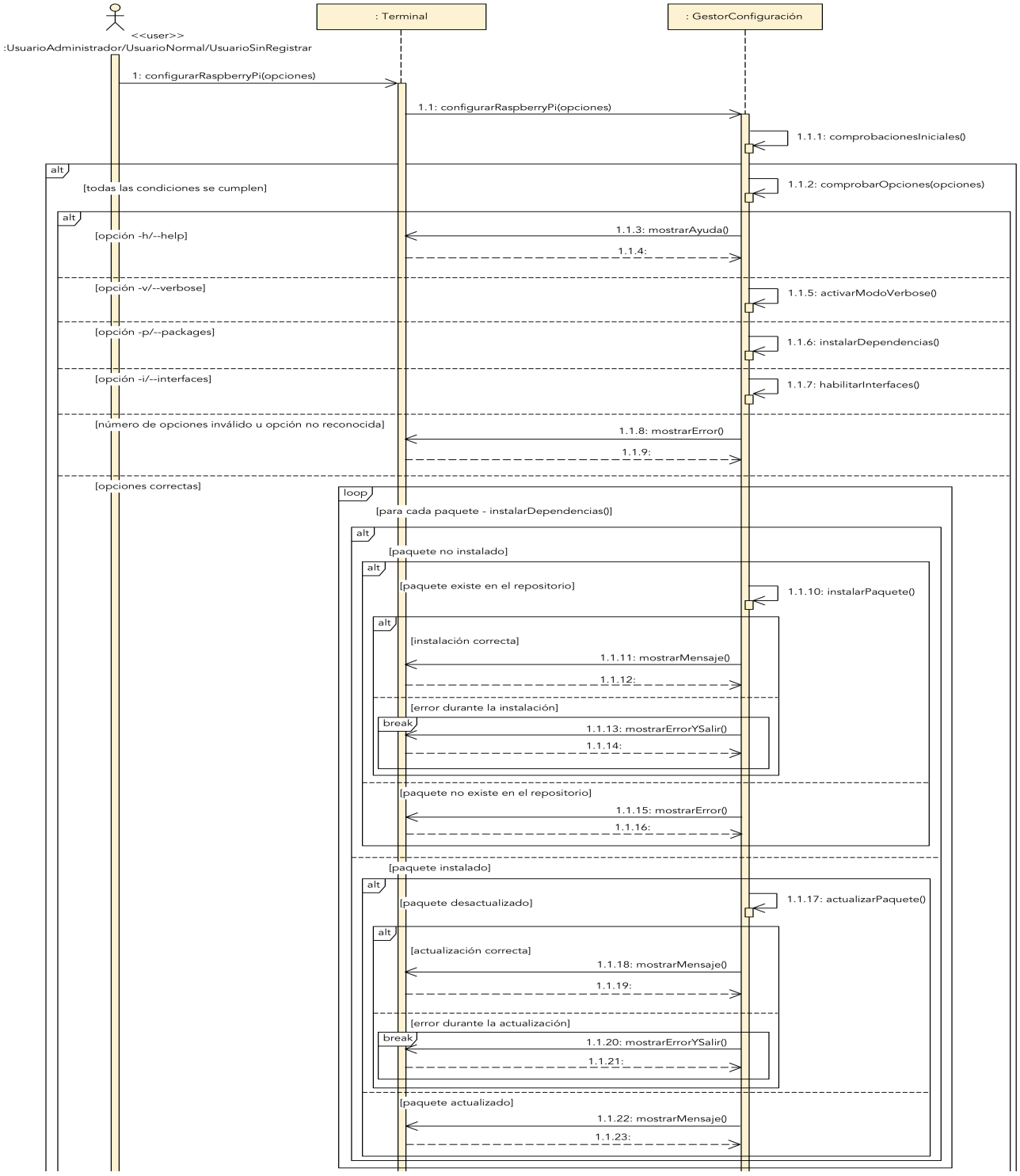
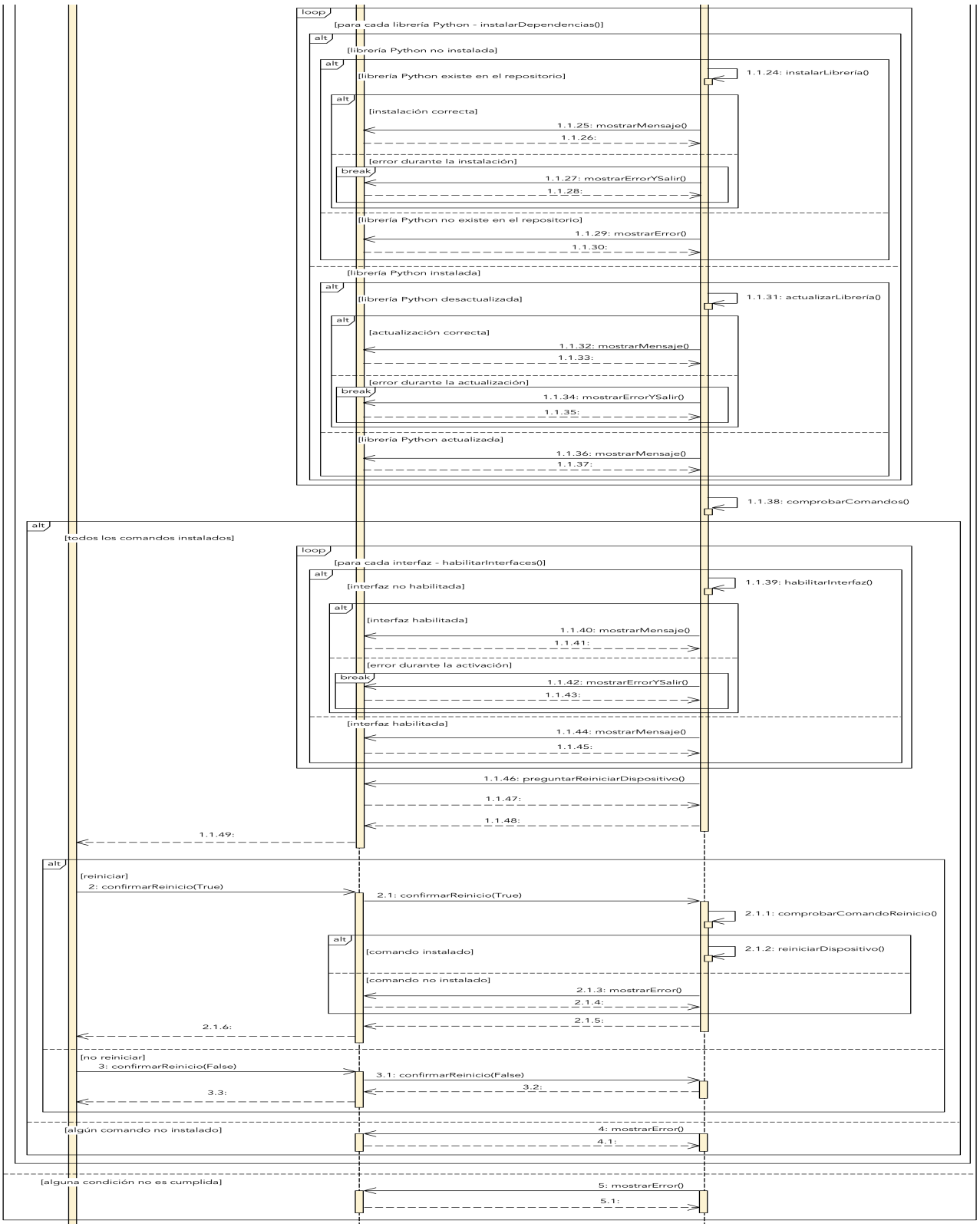
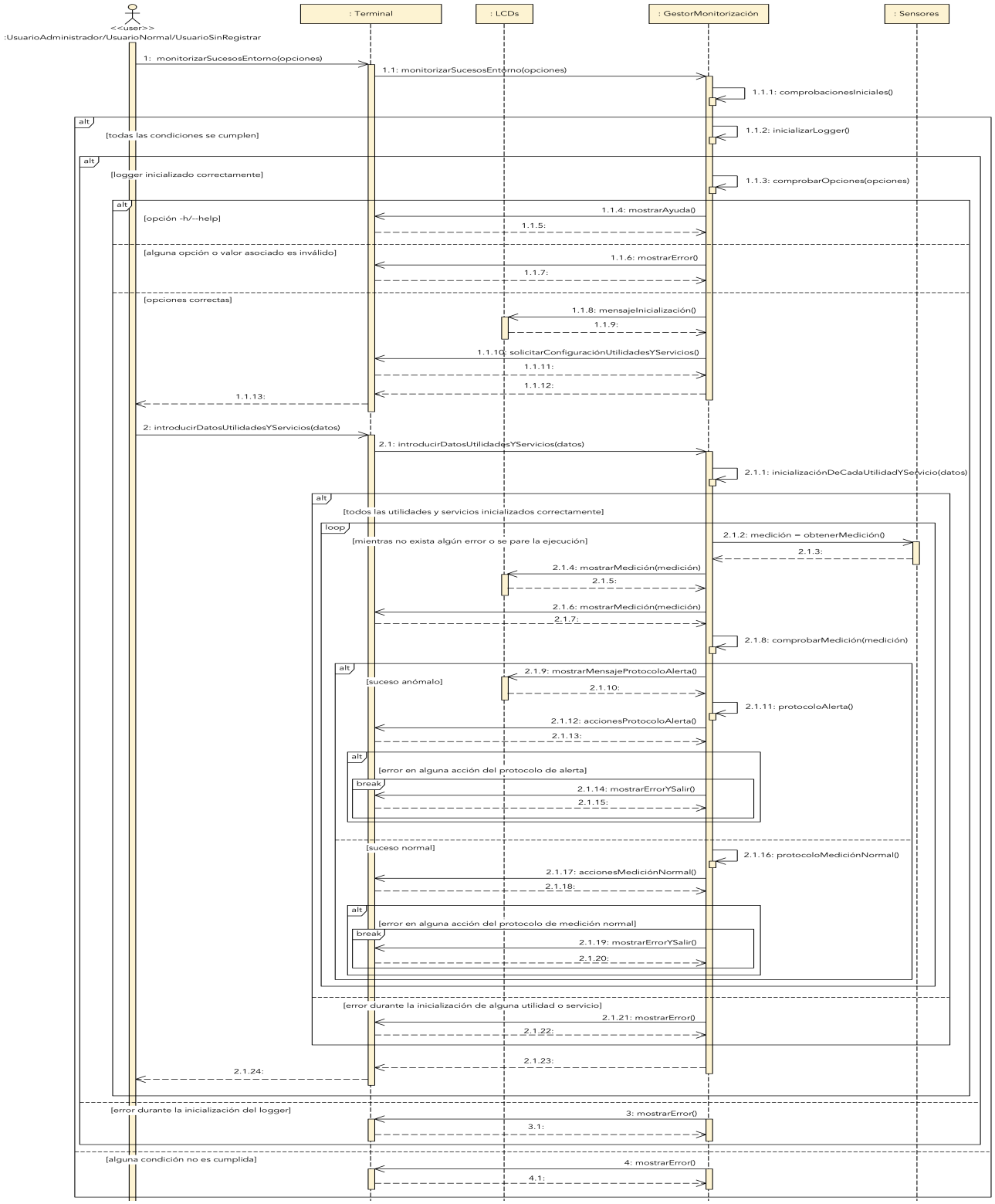


Figura 3.7: Diagrama de secuencia de análisis: Configurar dispositivo RPi - Parte 2.



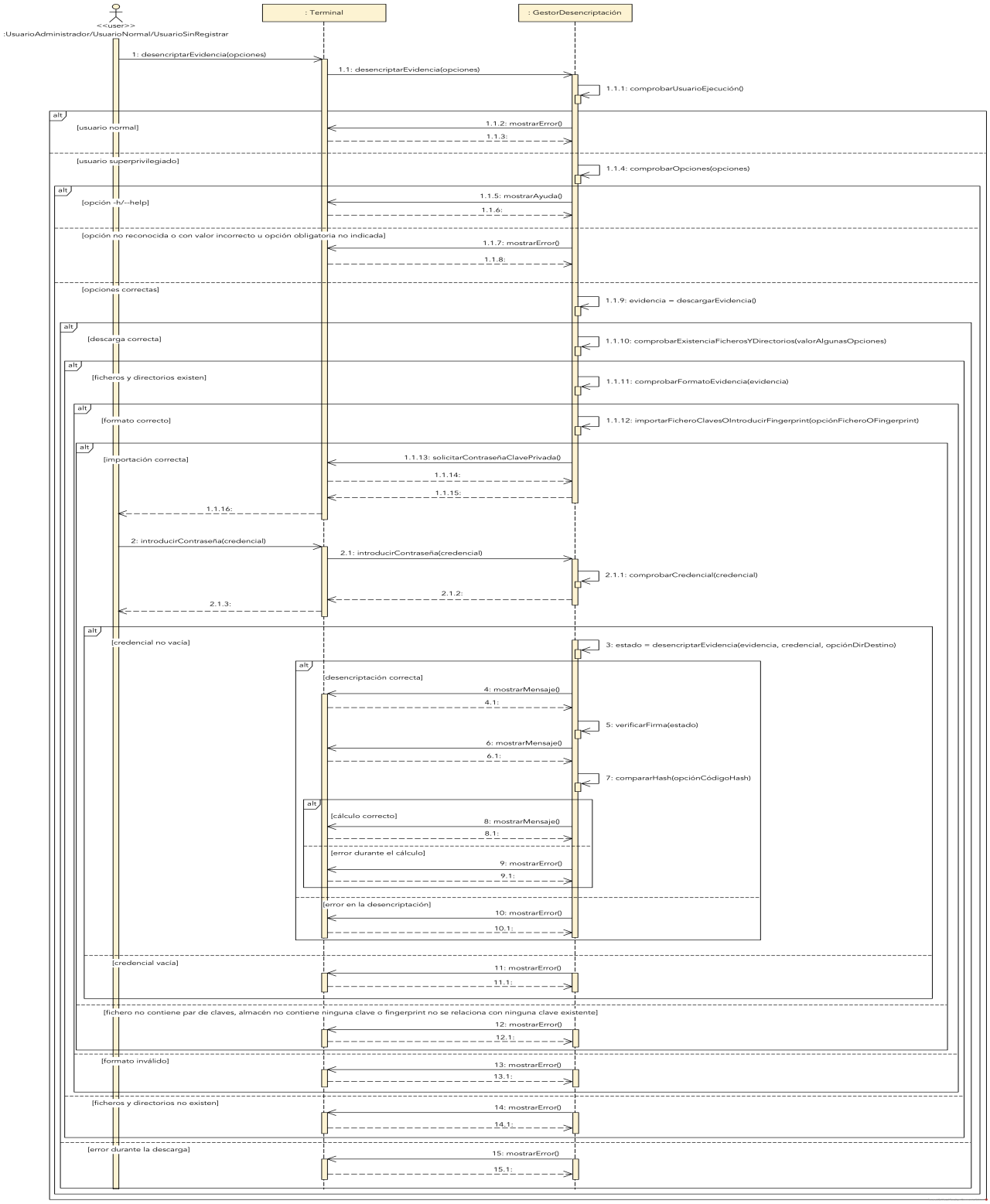
- Diagrama de secuencia de análisis: Monitorizar sucesos del entorno (Figura 3.8).

Figura 3.8: Diagrama de secuencia de análisis: Monitorizar sucesos del entorno.



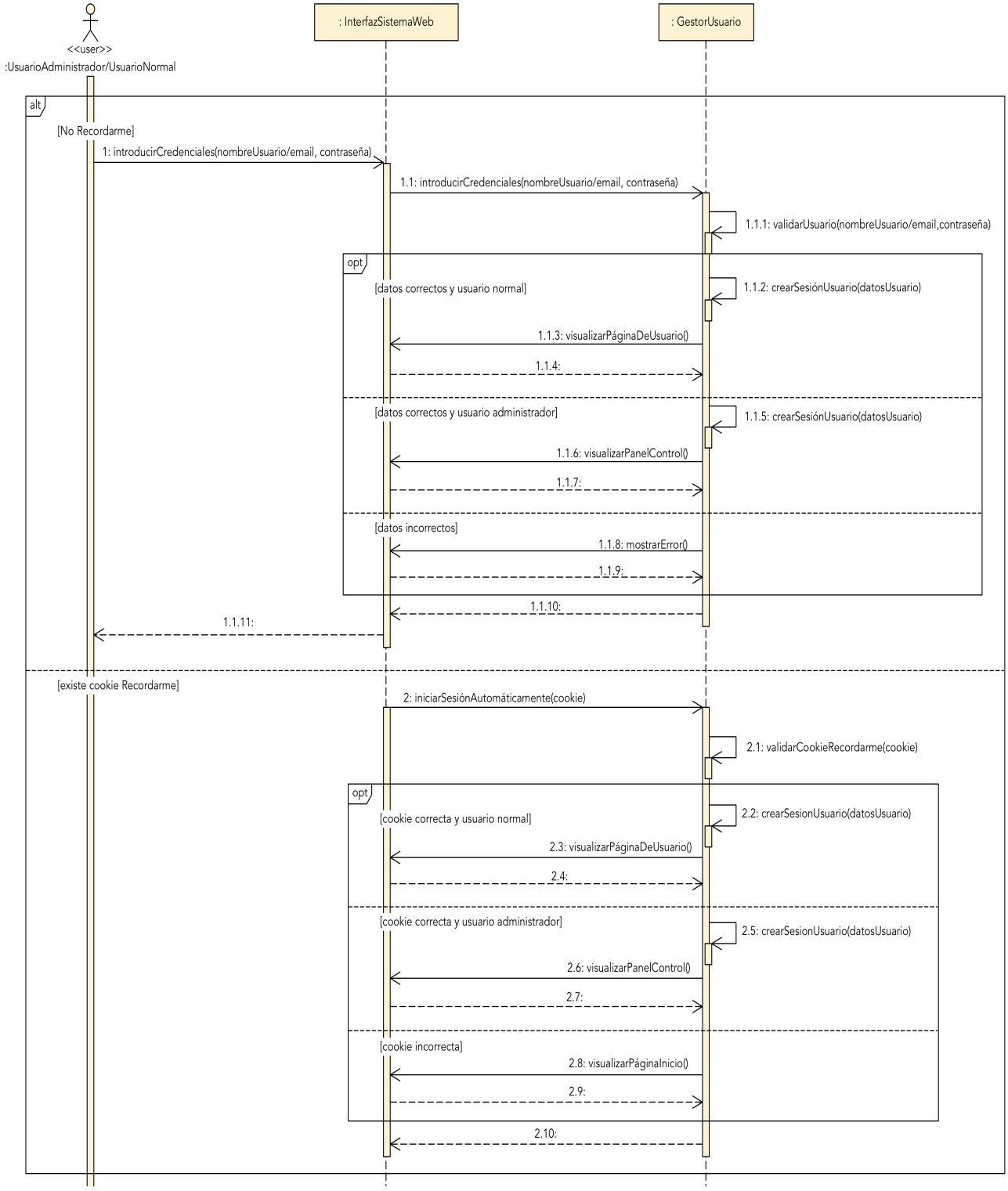
- Diagrama de secuencia de análisis: Desencriptar evidencia (Figura 3.9).

Figura 3.9: Diagrama de secuencia de análisis: Desencriptar evidencia.



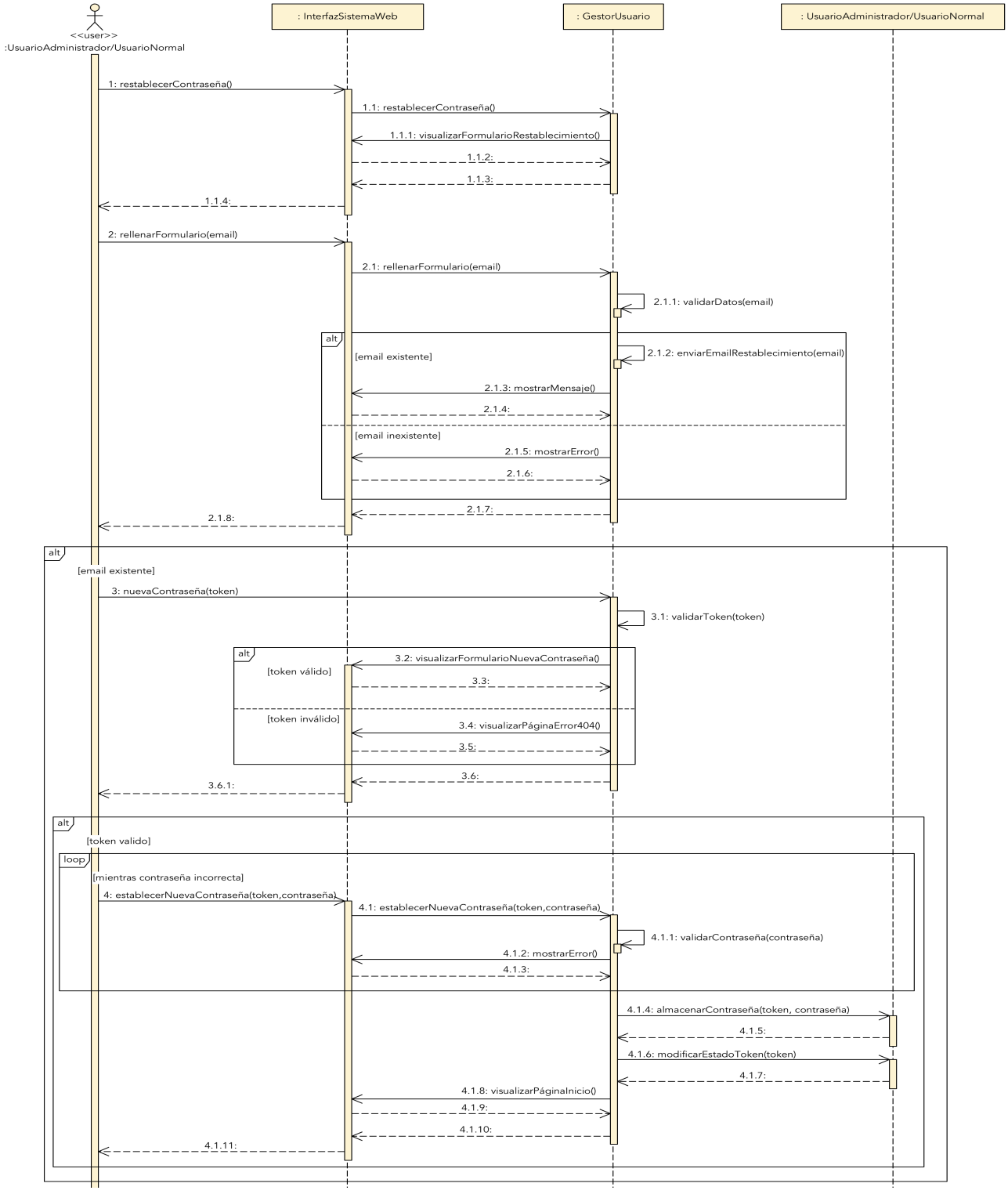
- Diagrama de secuencia de análisis: Iniciar sesión (Figura 3.10).

Figura 3.10: Diagrama de secuencia de análisis: Iniciar sesión.



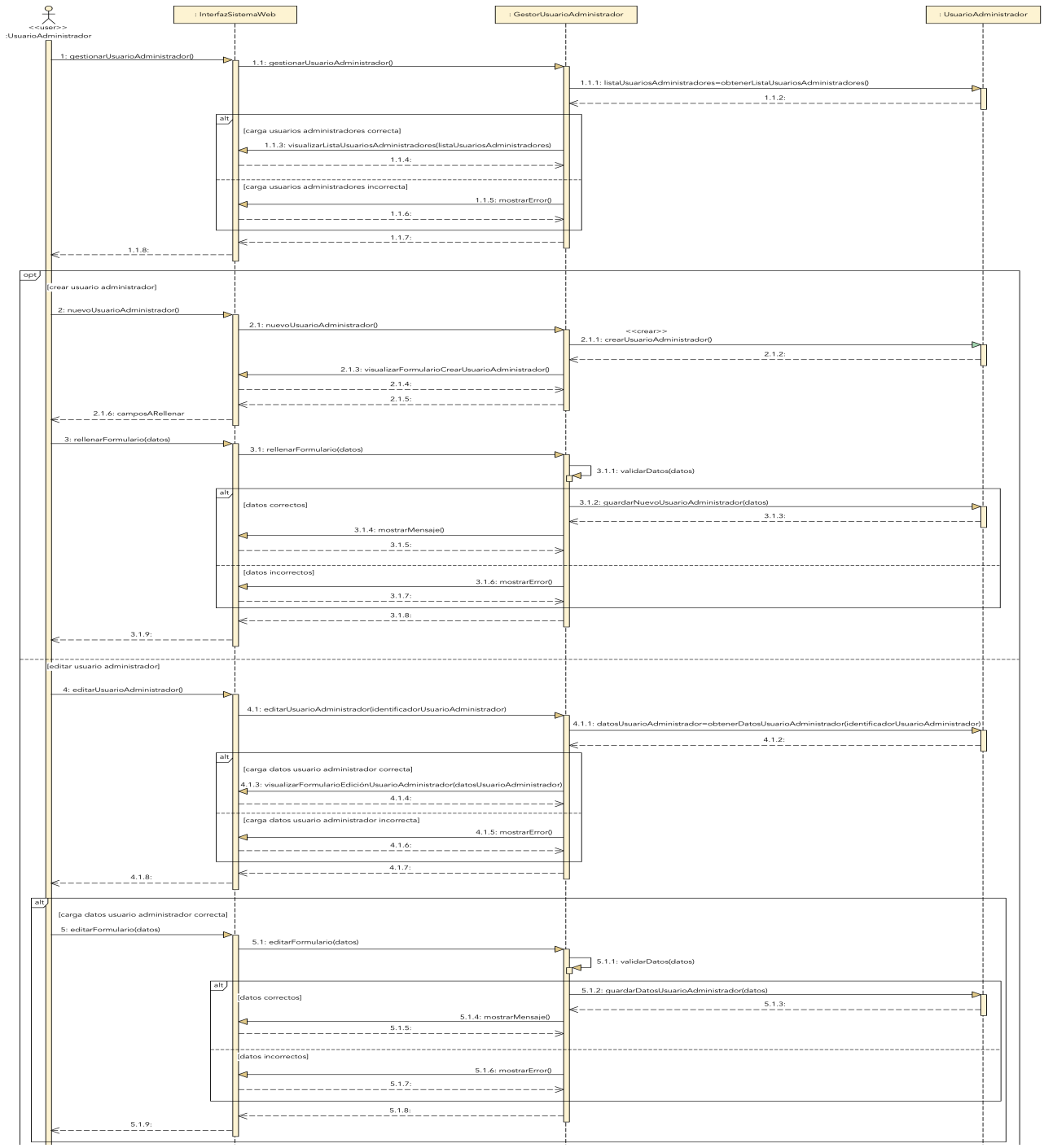
- Diagrama de secuencia de análisis: Restablecer contraseña (Figura 3.11).

Figura 3.11: Diagrama de secuencia de análisis: Restablecer contraseña.



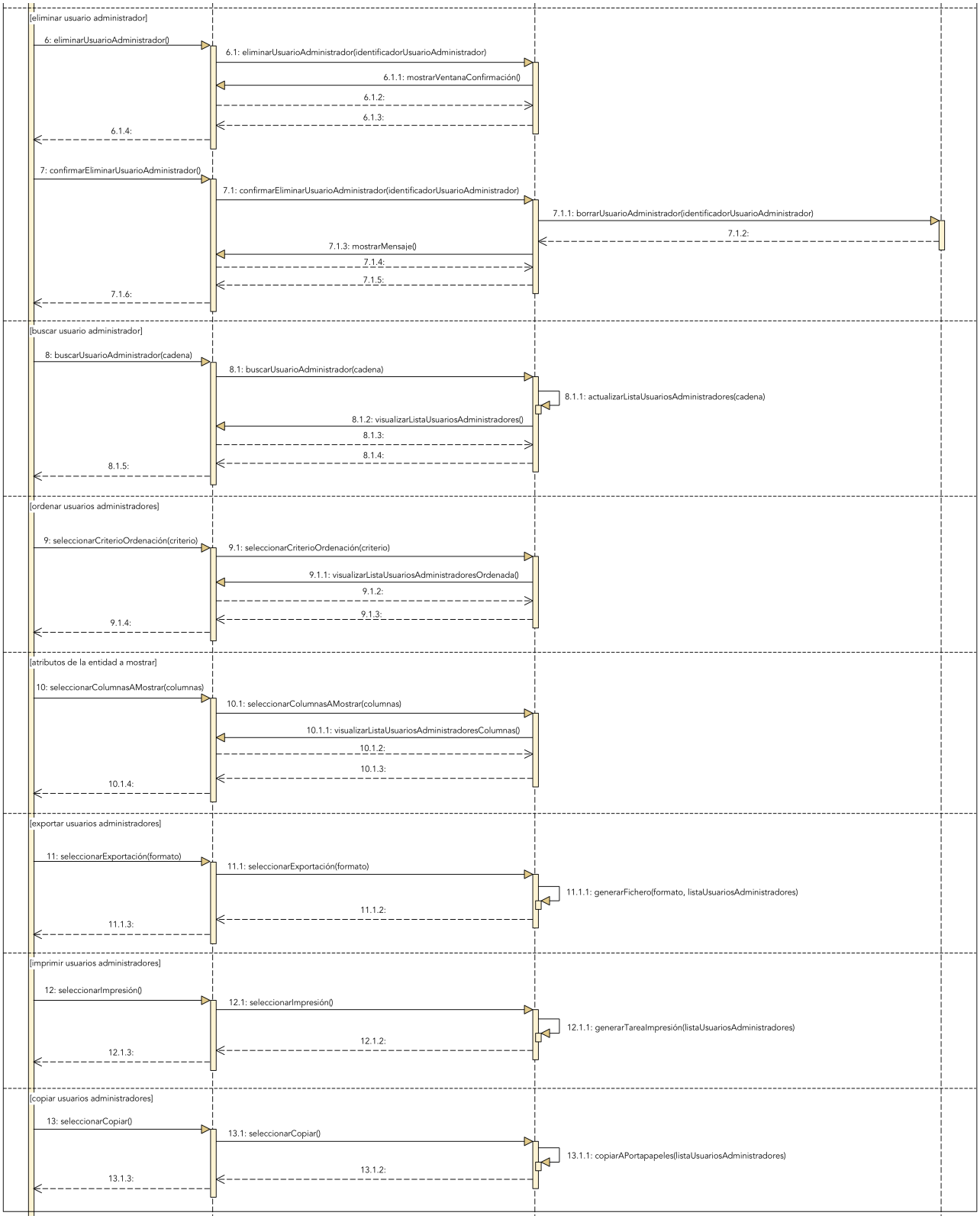
- Diagrama de secuencia de análisis: Gestionar usuario administrador (Figura 3.12 - Parte 1 y Figura 3.13 - Parte 2)[2].

Figura 3.12: Diagrama de secuencia de análisis: Gestionar usuario administrador - Parte 1.



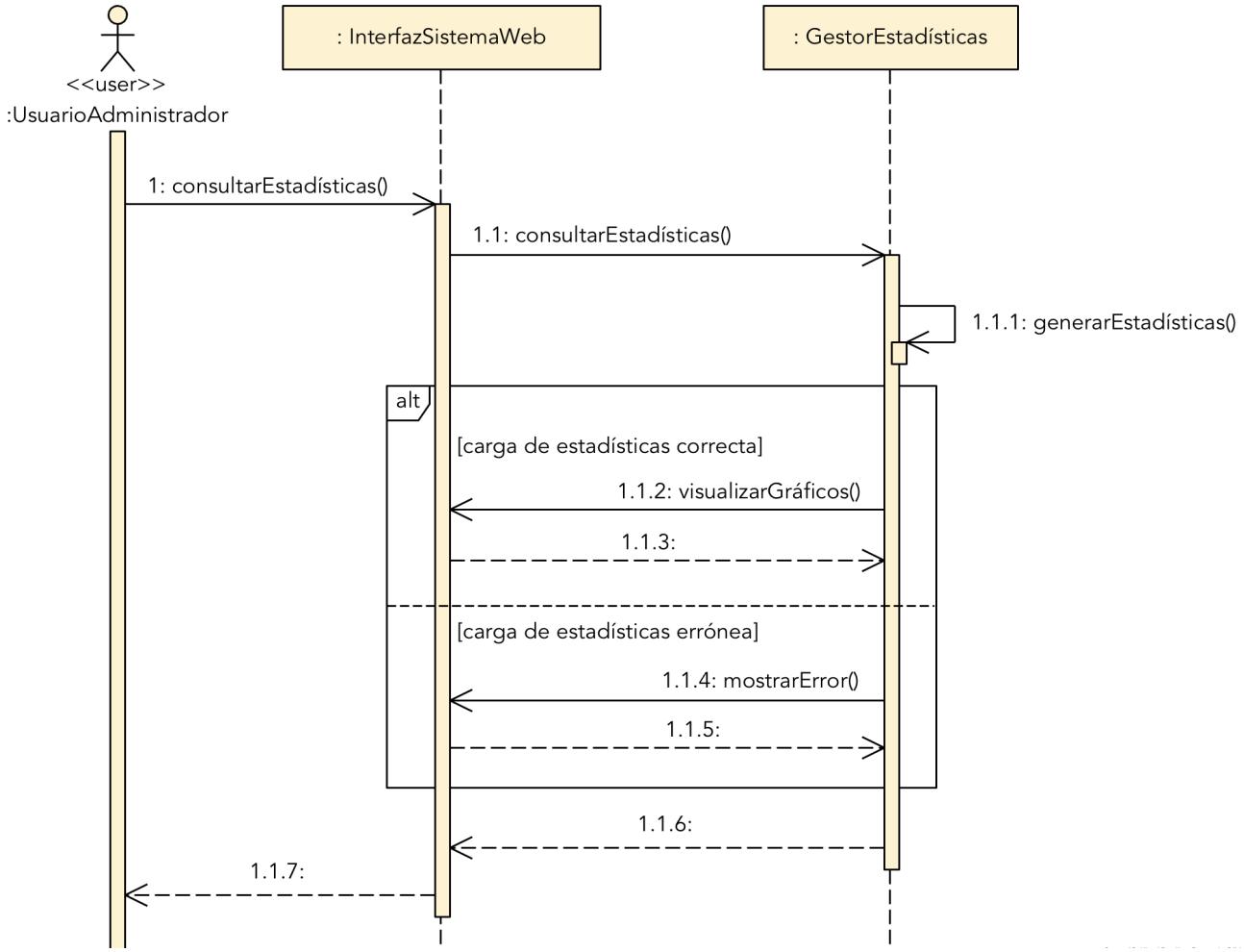
[2]El diagrama de secuencia de análisis para el caso de uso UC-0007: Gestionar usuario normal es similar al presente con la salvedad de adaptar a la entidad correspondiente.

Figura 3.13: Diagrama de secuencia de análisis: Gestionar usuario administrador - Parte 2.



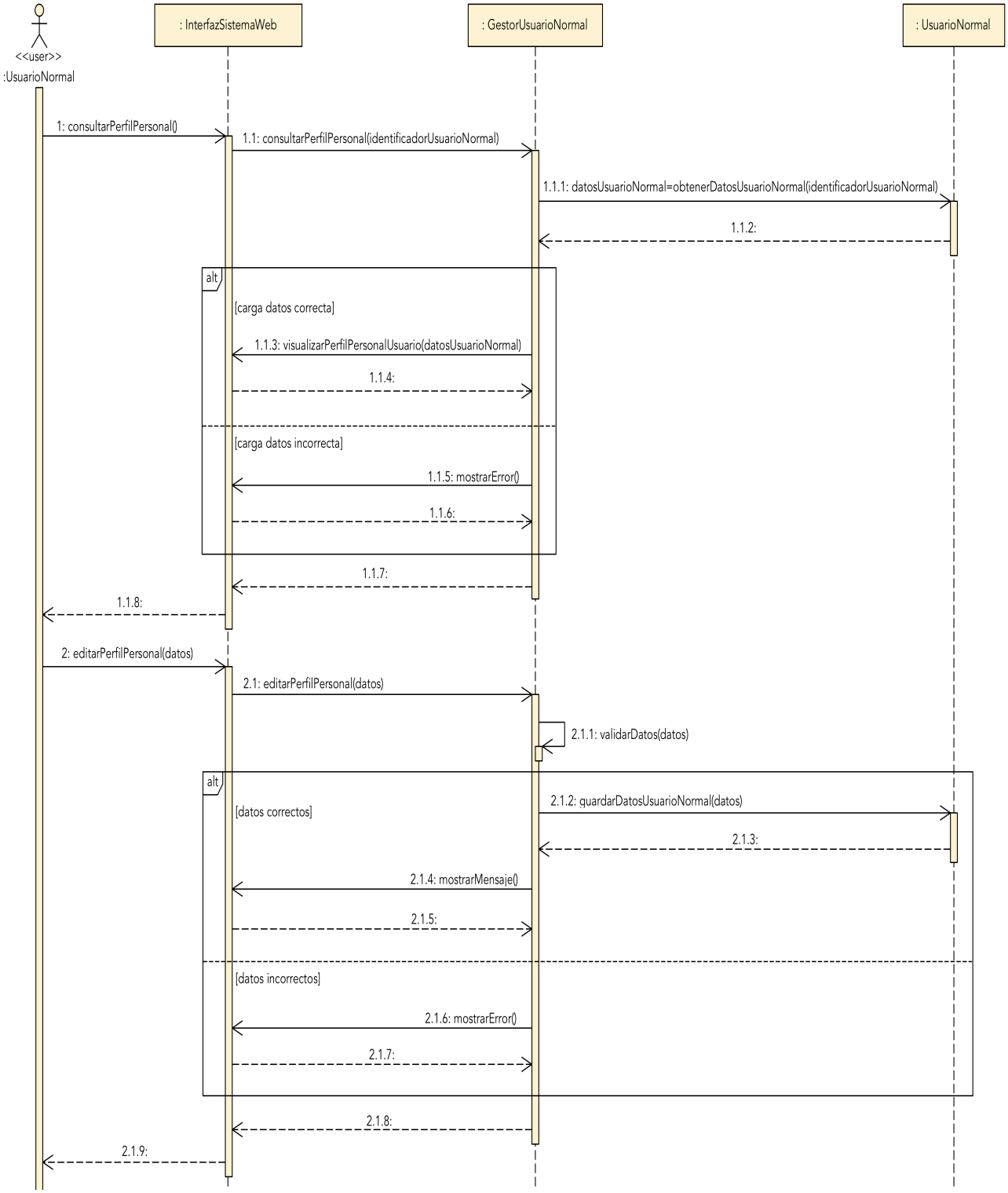
- Diagrama de secuencia de análisis: Consultar estadísticas globales (Figura 3.14).

Figura 3.14: Diagrama de secuencia de análisis: Consultar estadísticas globales.



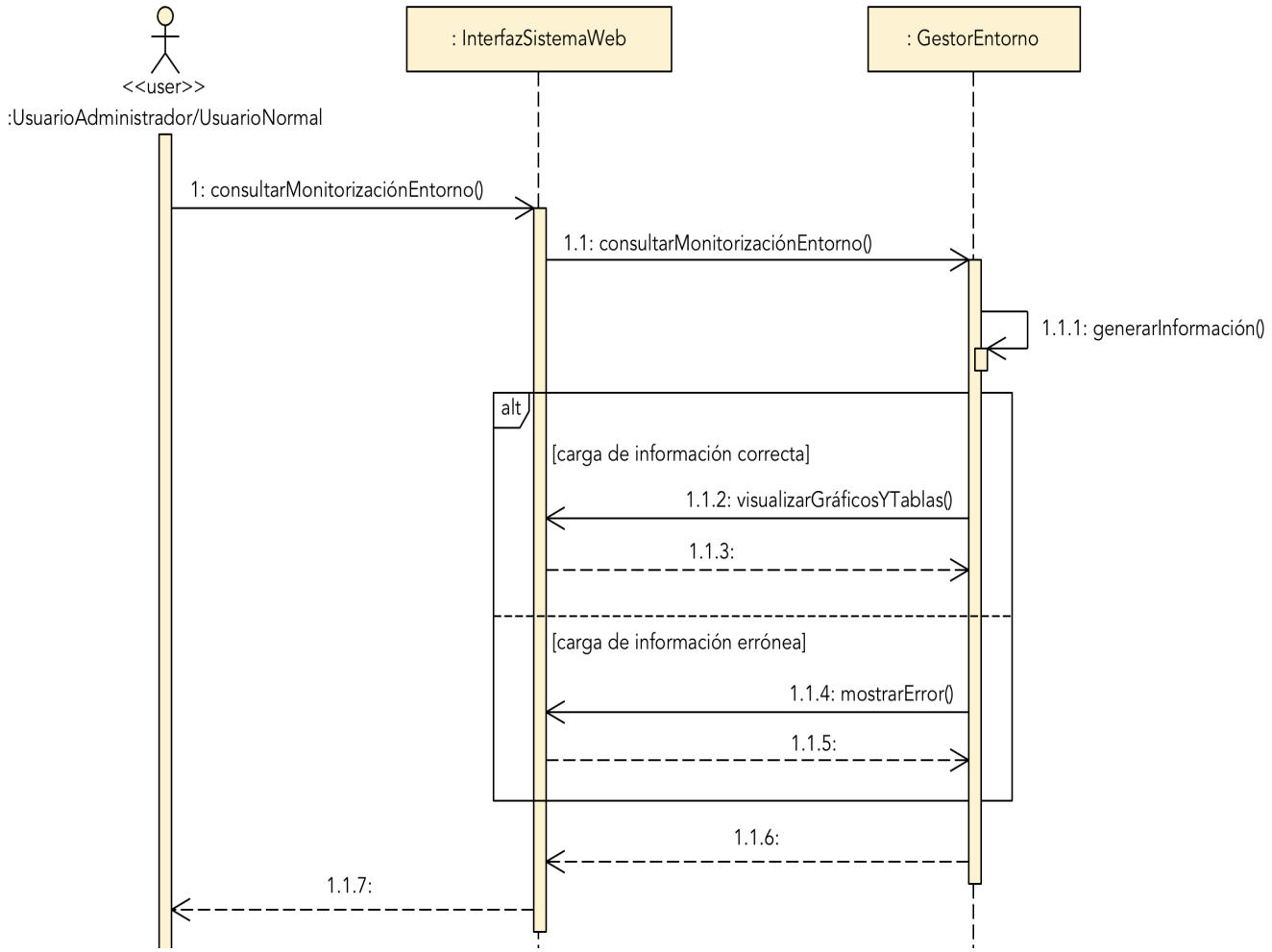
- Diagrama de secuencia de análisis: Consultar o modificar perfil personal (Figura 3.15).

Figura 3.15: Diagrama de secuencia de análisis: Consultar o modificar perfil personal.



- Diagrama de secuencia de análisis: Consumir información sobre la trazabilidad del entorno (Figura 3.16).

Figura 3.16: Diagrama de secuencia de análisis: Consumir información sobre la trazabilidad del entorno.



Capítulo 4

Diseño

Dotar de un diseño y una arquitectura al sistema tras haber recabado información durante la etapa de análisis, es esencial para formular y caracterizar la solución propuesta desde el punto de vista de diseño ya que acerca a los *stakeholders* a la implementación final. Este capítulo complementa la fase de análisis mostrando diferentes diagramas de diseño.

4.1. Modelado de la arquitectura física del sistema

En esta sección se realiza una descripción de la arquitectura *software* y *hardware* que guía la implementación del proyecto de trazabilidad en entornos IoT (*Internet of Things*) mediante Hyperledger, llamado **Hyot**. La finalidad es detallar con mayor énfasis cómo esta solución se encuentra construida, qué componentes incluye y cómo son las relaciones entre ellos. Para ello, se explican los patrones implementados explícitamente[1] y se muestran diferentes diagramas de diseño, elaborados con el lenguaje UML (*Unified Modeling Language*) [83], que permiten modelar el sistema tanto a nivel físico como lógico.

4.1.1. Descripción de la arquitectura

El diseño de la arquitectura del sistema tiene como objetivo dotar a la solución de diferentes patrones que resuelven problemas bien conocidos durante el desarrollo *software*, todo ello con el objetivo de favorecer la estructuración, mantenimiento y reutilización. En Hyot, se ha implementado en la medida de lo posible el patrón de diseño arquitectónico de *software* conocido como Modelo-Vista-Controlador (MVC) [8], aplicando la variante pasiva que se caracteriza porque únicamente el controlador manipula el modelo, es decir, la vista no tiene ninguna responsabilidad sobre la lógica de dominio y se encarga de tareas relacionadas con la interfaz de usuario (*User Interface* -UI-).

[1] Aquellos patrones implementados de forma implícita por las propias tecnologías no son mencionados (p.ej. patrón decorador o patrón de inyección de dependencias -DI-).

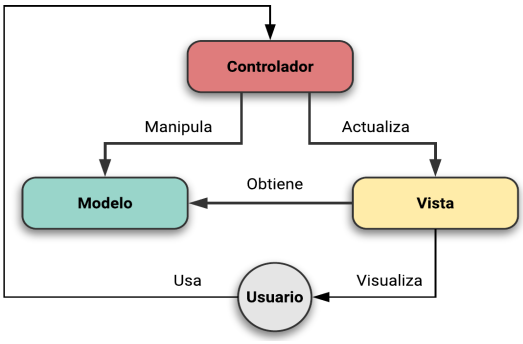
En el paradigma MVC, las entradas del usuario o lógica de negocio, los modelos del mundo exterior y la retroalimentación visual son explícitamente separados y manejados por tres tipos de objetos distintos, cada uno especializado para un conjunto de tareas específicas:

- Modelo: contiene una representación específica de la información que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Vista o UI: conforma la información que se envía al usuario y los mecanismos de interacción.
- Controlador: actúa como intermediario entre el modelo y la vista y como modificador del primero, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a lo solicitado por el usuario a través de la vista.

Este patrón de implementación sencilla y ampliamente utilizado, que se basa en la idea de favorecer el mantenimiento y reutilización de código, y de la separación de conceptos y características, busca facilitar la tarea de desarrollo. La Figura 4.1 muestra el esquema de este patrón en su variante pasiva, siendo el flujo de proceso habitual el especificado a continuación:

1. El usuario interactúa, de alguna forma, con la interfaz del sistema (interfaz web, terminal...).
2. El controlador recibe la entrada del usuario y la gestiona con un manejador de eventos.
3. El controlador accede y notifica al modelo la acción del usuario lo que puede implicar un cambio de estado de éste.
4. El controlador notifica a la vista de que el modelo ha sido modificado y delega a los objetos de ésta la tarea de actualizar la información de la interfaz.
5. La interfaz del sistema espera nuevas interacciones del usuario, comenzando el ciclo de nuevo.

Figura 4.1: Esquema del patrón arquitectónico MVC pasivo.



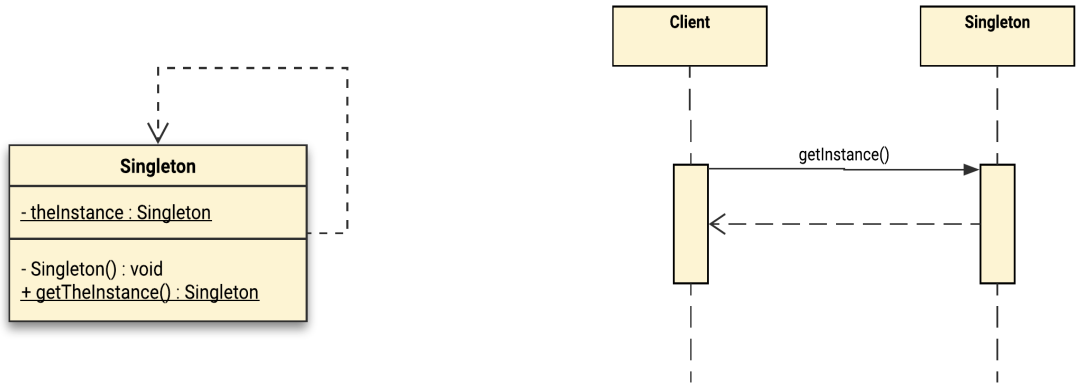
A parte de este patrón, también se ha implementado el patrón Singleton [23]. Este patrón es un patrón de diseño creacional ampliamente conocido y utilizado en el desarrollo *software* que está diseñado para garantizar la existencia de una única instancia de un objeto en toda la aplicación proporcionando así un punto único de acceso global. En sí, no se encarga de la creación de objetos sino de aplicar esta restricción durante la creación lo que proporciona los siguientes beneficios: acceso estricto a la instancia única, espacio de nombres reducido al no reservar nombres para variables globales, mayor número de variables de instancia y mayor flexibilidad. Su funcionamiento es sencillo y se reduce a (Figura 4.2):

1. Privatizar el constructor de la clase Singleton, para que los usuarios no puedan crear instancias.

2. Declarar en la clase Singleton una variable miembro privada que contenga la referencia a la instancia única que se quiere gestionar.

3. Proveer en la clase Singleton una función o propiedad que brinde acceso a la única instancia gestionada. Los usuarios acceden a la instancia a través de esta función o propiedad.

Figura 4.2: Diagrama de clases y de secuencia del patrón de diseño creacional Singleton.



Este patrón ha sido empleado en el componente de monitorización de sucesos del entorno, en concreto en:

- El *logger* para controlar el número de instancias que se generan de esta clase.

- En las importaciones de módulos. Por naturaleza, todos los módulos en Python se comportan como un patrón Singleton ya que se comprueba si el módulo ya ha sido importado con anterioridad. En caso afirmativo, lo retorna. En caso contrario, busca el módulo, lo inicializa y lo retorna por lo que únicamente se inicializa una vez definiendo un nombre o nombres en el espacio de nombres local.

4.1.2. Diagrama de capas

El diagrama de capas permite describir la arquitectura de la solución a alto nivel segmentando los elementos en grupos abstractos desde el punto de vista lógico, denominados capas -nodos en el diagrama-. Cada una puede describir las tareas principales que realizan los componentes del sistema, limitando el flujo de comunicación solamente a capas contiguas. El objetivo primordial de este modelo es el desacoplamiento de las diferentes partes que componen la solución *software*, de forma que por ejemplo la creación de un nuevo componente visual o su modificación se pueda simplificar de gran manera al no requerir modificación alguna en el resto de capas, salvo la afectada lo que implica una reducción del acoplamiento entre componentes.

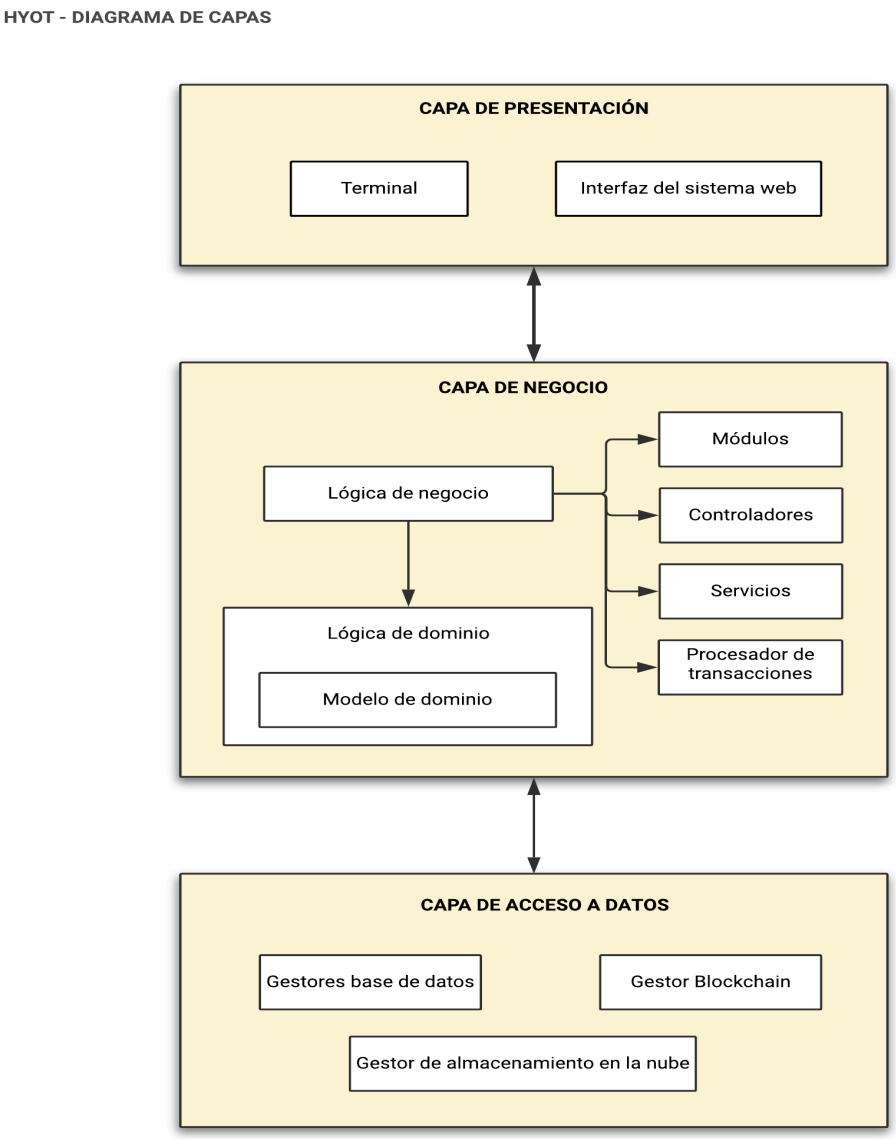
La división se ha realizado en 3 capas (Figura 4.3), cada una de ellas con una misión, favoreciendo así un diseño modular y escalable. A continuación, se describe cada capa identificada:

- Capa de presentación o de usuario: capa que presenta el sistema al usuario comunicándole la información y permitiéndole interactuar con el propio sistema a través de la captura de la información introducida por el usuario. Esta capa se comunica únicamente con la capa de negocio y se corresponde con el componente vista en el patrón MVC. Debido a que se trata del nivel que expone y visualiza la información al usuario, debe implementar una UI simple y fácil de usar que presente claridad y coherencia en los textos y elementos, etc. Esta capa está representada por el terminal o consola y por la interfaz del sistema web.

- Capa de negocio o aplicativa: capa que contiene la lógica de negocio de la solución la cual está compuesta por una serie de reglas que se aplican sobre las peticiones del usuario para originar las respuestas que serán retornadas tras este proceso. Esta capa se comunica con la capa de presentación para recibir solicitudes y presentar los resultados y con la capa de datos para solicitar a los mecanismos de persistencia las acciones de almacenar, modificar o recuperar información. En el paradigma MVC, esta capa se corresponde con el componente controlador y haciendo referencia al proyecto, la lógica de negocio comprende toda aquella codificación, incluyendo módulos, clases, controladores, servicios, procesador de transacciones (*transactions*), etc., que trata la información introducida por el usuario, ya sea a través del terminal o de la interfaz del sistema web o incluso aquella información que es introducida al sistema de manera automática procedente de fuentes de información externa como son los sensores, y genera respuestas.

- Capa de datos o de persistencia: representa el modelo, es decir, la información almacenada y tratada. Esta capa, formada por la información almacenada en los cuatro mecanismos de persistencia empleados, es la responsable de la gestión y almacenamiento permanente de los datos según las peticiones recibidas de la capa de negocio, correspondiendo al modelo en el paradigma MVC.

Figura 4.3: Diagrama de capas.



4.1.3. Diagrama de despliegue

El diagrama de despliegue modela la arquitectura en tiempo de ejecución de un sistema, es decir, muestra la topología del sistema, la estructura de los componentes *hardware*, los componentes *software* que se ejecutan en cada nodo y la forma en la que las distintas partes están conectadas entre sí.

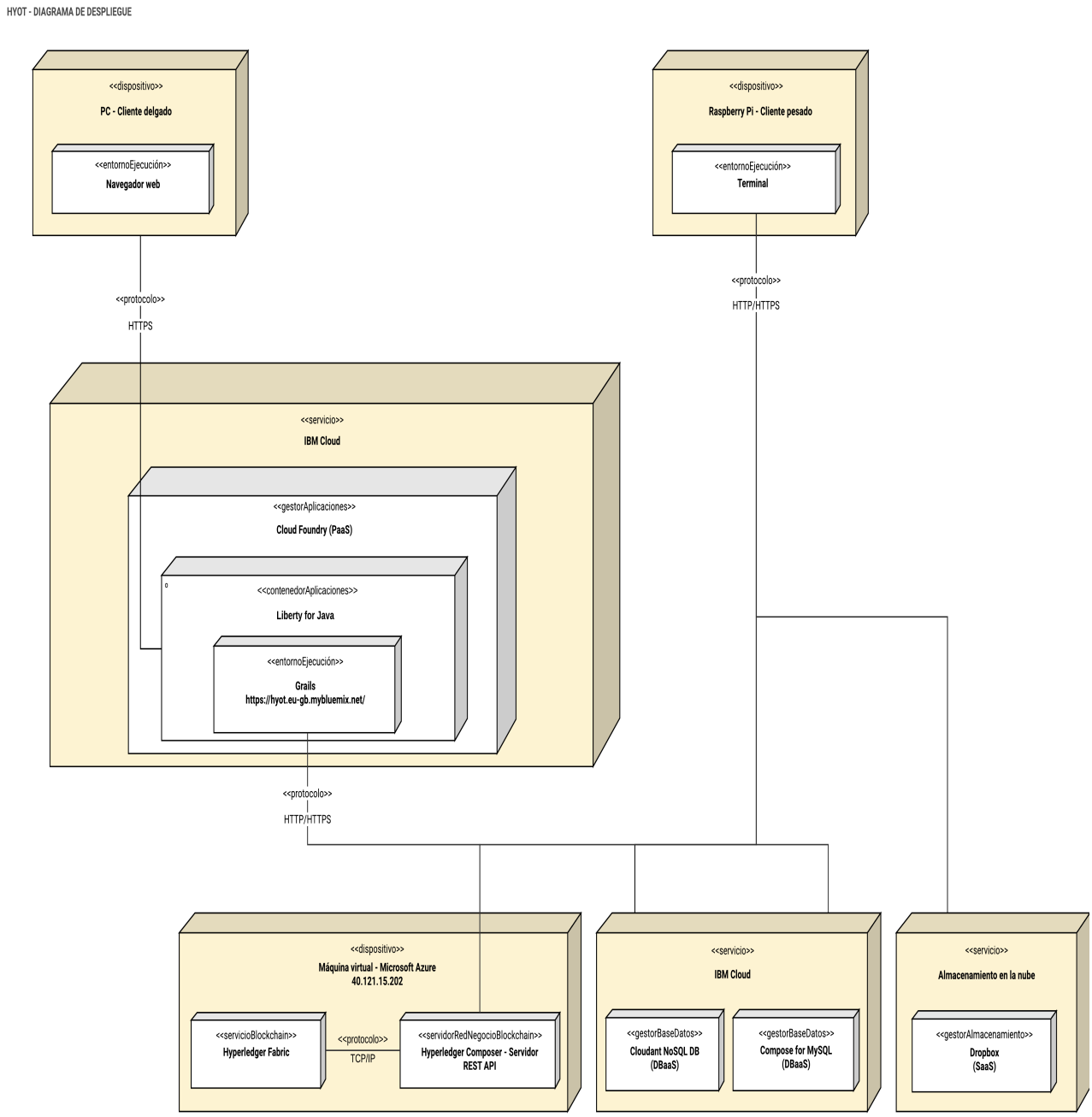
En Hyot, se ha optado por implementar una arquitectura centralizada basada en el paradigma cliente-servidor y en el modelo *3-tier* (3 niveles) lo que significa que las capas lógicas se encuentran distribuidas de forma física en 3 componentes distintos. Este modelo es una extensión del modelo

tradicional en 2 niveles que pretende aumentar el desacomplamiento entre servidor/es y clientes a través de la introducción de un nivel intermedio para separar el servidor en dos componentes: servidor que contiene la lógica de negocio y servidores que contienen los mecanismos de persistencia. Además, este modelo presenta mayor flexibilidad y modularidad, escalabilidad, extensibilidad, seguridad, reusabilidad de componentes, aislamiento frente a cambios en otras capas e independencia frente a cambios en los componentes de persistencia.

El motivo de esta elección se debe a que se trata de la arquitectura que mejor concuerda con el sistema debido a la existencia de un conjunto de clientes que demandan un servicio proporcionado por el conjunto de servidores de la solución. Además, la transferencia de información entre éstos se puede optimizar al recudir la carga de trabajo y el número de peticiones de cada uno de ellos y de utilizar protocolos de comunicación de bajo nivel muy rápidos reduciendo el tráfico de red. Cabe indicar que esta arquitectura favorecerá el rendimiento en situaciones de saturación donde la carga solicitada será mayor que en el contexto y entorno actual de prueba. Desglosando la arquitectura mostrada en el diagrama de despliegue (Figura 4.4), se pueden observar los siguientes niveles:

- Primer nivel: en este primer nivel se ubica la capa de presentación donde se diferencian dos tipos de dispositivos *hardware*.
    - Dispositivo que representa un cliente delgado ya que simplemente actúa como intermediario entre el usuario y el sistema web, no implementando ningún aspecto de la lógica de negocio. Este dispositivo puede ser cualquier ordenador o dispositivo móvil con conexión a Internet el cual permite a los usuarios interactuar con el sistema web mediante un navegador utilizando el protocolo HTTPS (*Hypertext Transfer Protocol Secure*).
    - Dispositivo Raspberry Pi (RPi) que representa un cliente pesado o grueso ya que delega carga de cómputo y capacidad de procesamiento de datos en el propio dispositivo, gracias a la ejecución de un componente del proyecto. Este dispositivo interactúa directamente mediante los protocolos HTTP (*Hypertext Transfer Protocol*) y HTTPS con los mecanismos de persistencia desplegados.
- Segundo nivel: en este nivel se localiza el contenedor de aplicaciones (*Liberty for Java*) donde se despliega el sistema web en la plataforma como servicio (*Platform as a Service* -PaaS-) [61] Cloud Foundry de IBM Cloud. Este sistema web interactúa directamente mediante los protocolos HTTP y HTTPS con los mecanismos de persistencia desplegados.
- Tercer nivel: en el último nivel se hallan los cuatro mecanismos de persistencia utilizados, el servicio de almacenamiento en la nube, los gestores de bases de datos desplegados en el servicio IBM Cloud y el servicio de Blockchain (BC) de Hyperledger Fabric (HF) desplegado en una máquina virtual (*Virtual Machine* -VM-) en Microsoft Azure.

Figura 4.4: Diagrama de despliegue.

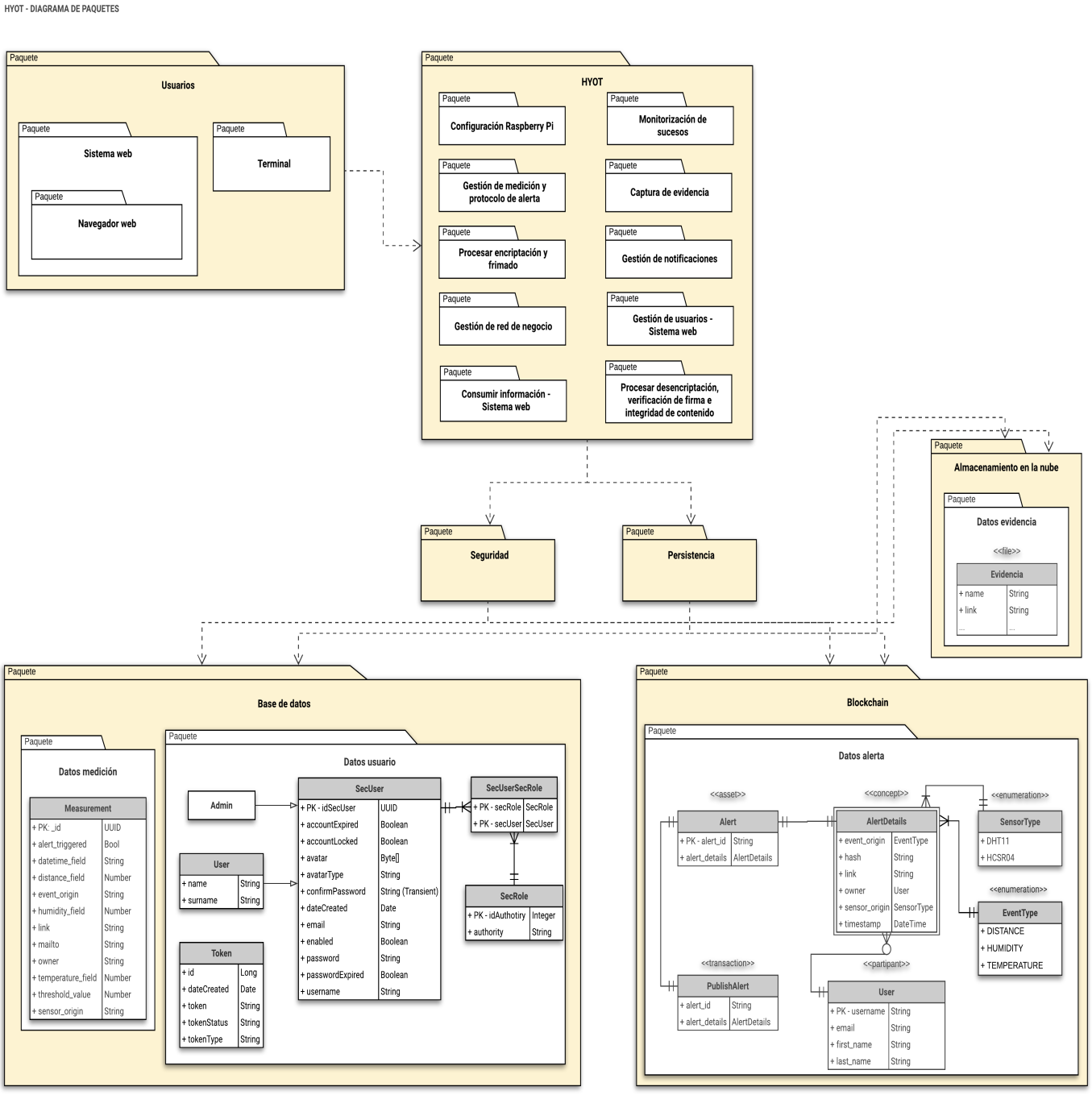


4.1.4. Diagrama de paquetes

El diagrama de paquetes muestra el sistema dividido en agrupaciones lógicas individuales exponiendo las dependencias existentes entre ellas y proporcionando una descomposición de la jerarquía lógica. Los paquetes, mecanismo de agrupación de elementos modelados con UML, faci-

litan el manejo de los modelos en un sistema complejo definiendo un espacio de nombres y están normalmente organizados para maximizar la coherencia interna dentro de cada paquete y minimizar el acoplamiento externo entre los paquetes. La Figura 4.5 muestra el diagrama de paquetes definido:

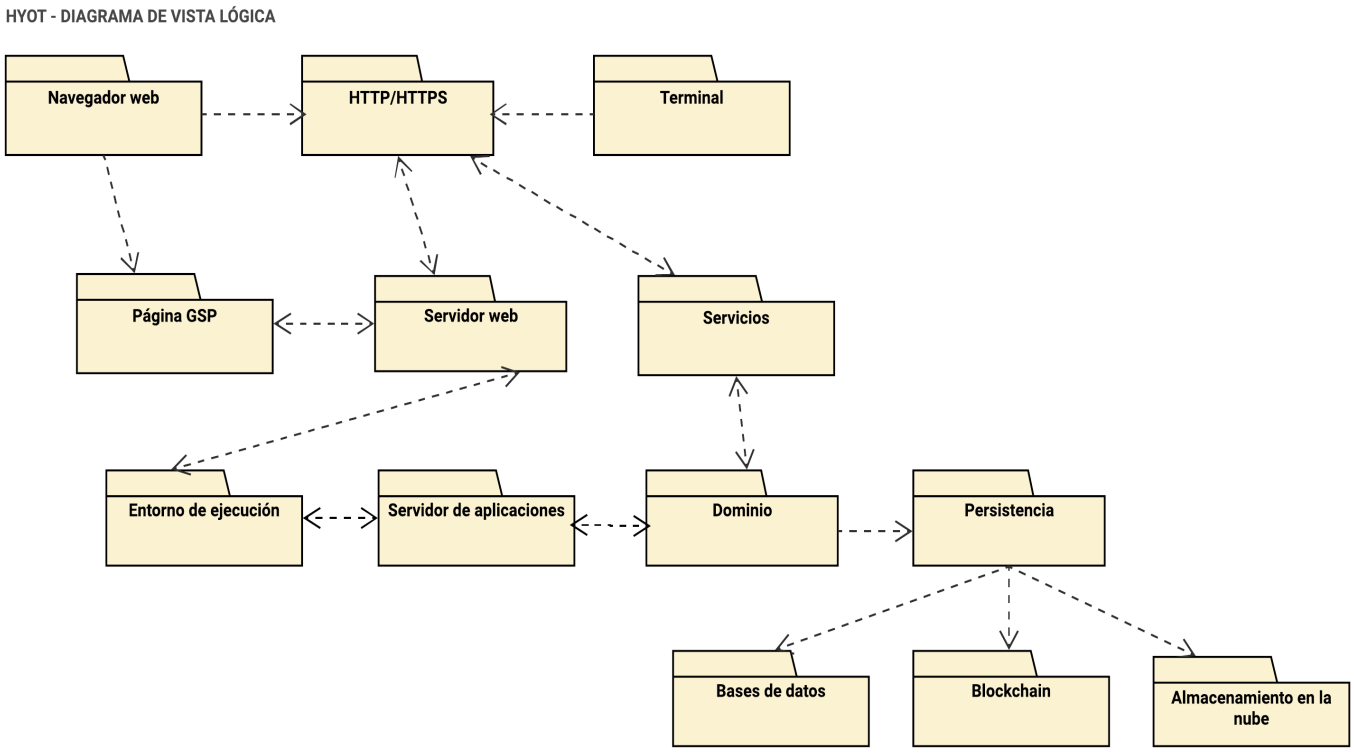
Figura 4.5: Diagrama de paquetes.



4.1.5. Diagrama de vista lógica

El diagrama de vista lógica es una abstracción a gran escala del modelado de paquetes, subsistemas y clases, buscando mostrar los requerimientos funcionales del sistema y cómo éste se debe comportar y funcionar. La Figura 4.6 muestra el diagrama definido:

Figura 4.6: Diagrama de vista lógica.



4.1.6. Capa de persistencia

La capa de persistencia es una capa independiente de la capa de negocio que se encarga de abstraer, encapsular y resolver completamente el acceso a los datos por parte de las aplicaciones cliente que la utilicen, siendo posible trabajar con varios mecanismos de persistencia ya sean gestores de bases de datos u otros medios o servicios que permitan almacenar información. Su objetivo es ser la única capa que conoce cómo son persistidos los objetos de dominio de la aplicación y cómo pueden ser recuperados abstrayendo el choque de impedancias entre objetos y tablas relacionales o no relacionales. Las capas superiores no interactúan directamente con estos mecanismos, sino que lo hacen mediante la interfaz expuesta por la capa de persistencia logrando así la independencia buscada. Ésto proporciona la principal ventaja de las capas de abstracción, la portabilidad debido a que es posible cambiar la estrategia en la que los objetos son persistidos e incluso modificar

la tecnología o el motor utilizado, sin impactar al código restante que se ubica en capas superiores.

En el proyecto se emplean cuatro mecanismos de persistencia para la información manipulada, cada uno de ellos con una finalidad distinta:

- Base de datos (BBDD) distribuida como servicio (*Database as a Service* -DBaaS-) [61] en la nube de tipo no relacional (NoSQL) y orientada a documentos en formato JSON (*JavaScript Object Notation*) que almacena la información de cada medición obtenida. La decisión de su elección se tomó en base a las necesidades de escalabilidad y versatilidad de la información almacenada puesto que la principal característica de este tipo de BBDD es que se encuentran optimizadas para grandes volúmenes de datos con un esquema flexible lo que se alinea con el contexto del proyecto, mediciones constantes y posibilidad de ampliación del proyecto a otros dispositivos IoT lo que ampliaría o modificaría el modelo de información a almacenar. Además, la elección de ser distribuida como un servicio y no usar un despliegue local se debe a facilitar el proceso de puesta en marcha y configuración y la disponibilidad total ya que se encuentra físicamente remota pero lógicamente en local.
- BBDD de tipo relacional (SQL) y DBaaS que almacena la información que gestiona el sistema web sobre usuarios y *tokens* para proporcionar autenticación y restringir así el libre acceso a la información monitorizada. Debido a la tecnología utilizada, se emplea implícitamente una herramienta de mapeo objeto-relacional (*Grails object relational mapping* -GORM-) construida sobre Hibernate y escrita en Groovy para mapear el modelo a tablas de la base de datos y controlar el ciclo de vida de las entidades. La decisión de su elección se tomó en base a la necesidad de requerir combinar de forma eficiente diferentes tablas para extraer información relacionada sin importar la posible modificación de la estructura de los datos o crecimiento del volumen de éstos puesto que esta información se encuentra bien definida y acotada y no es necesario las ventajas de una BBDD NoSQL.
- Servicio de almacenamiento en la nube para almacenar los artefactos -evidencias- encriptados y firmados que son generados al al monitorizar los sucesos del entorno. Debido a su tamaño y tipología lo más adecuado es su almacenamiento en un servicio en la nube donde estén siempre disponibles, siempre y cuando este almacenamiento se realice con algún tipo de seguridad.
- Blockchain (BC): mecanismo de persistencia principal donde se almacena aquella información que se quiera salvaguardar de cualquier posible alteración por una parte no autorizada y cuyo almacenaje en una BBDD no proporciona la seguridad requerida.

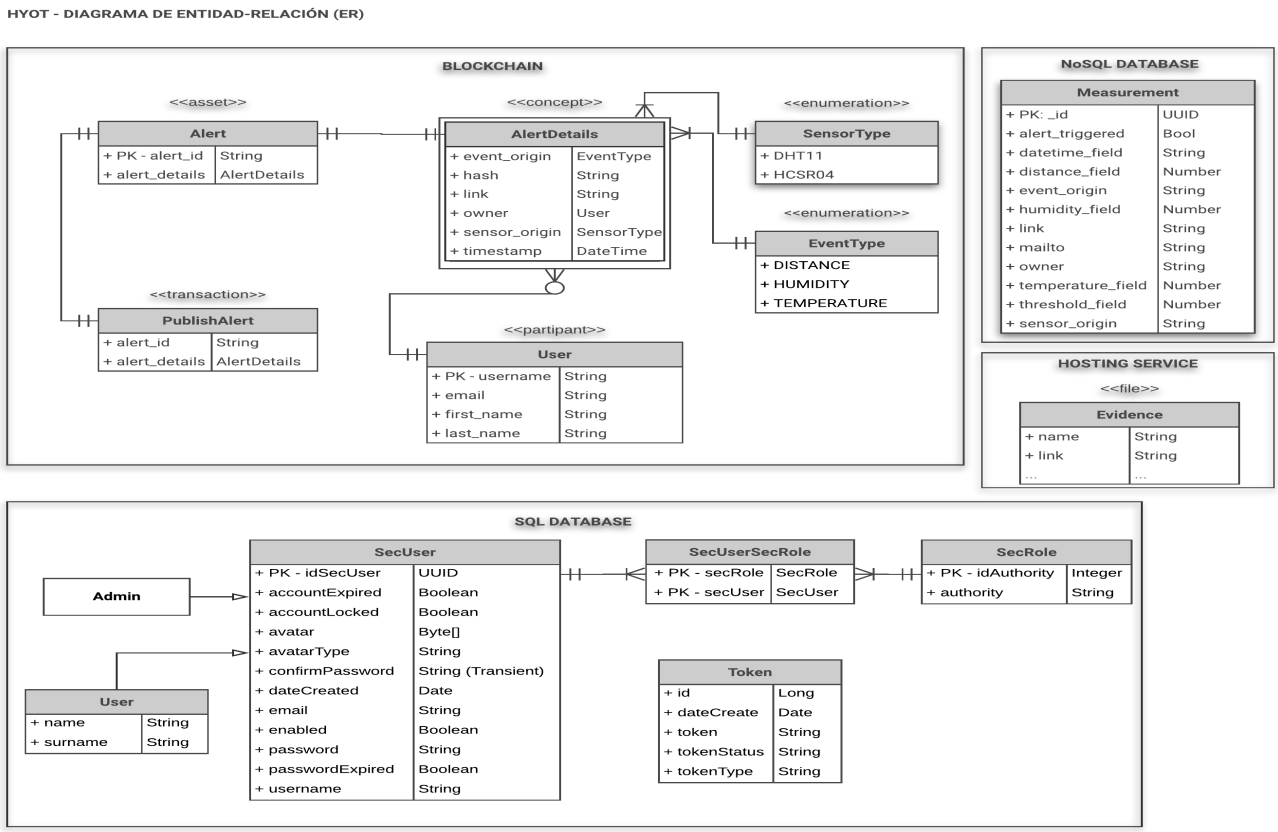
Diagrama Entidad-Relación

El diagrama Entidad-Relación es un modelo de datos basado en una percepción del mundo real que consiste en un conjunto de objetos básicos llamados entidades y relaciones entre ellos para

representar las entidades relevantes de un sistema así como sus interrelaciones y propiedades. Se puede obtener con la técnica, ingeniería inversa la cual permite conseguir información y su diseño a partir de un sistema y por tanto poder determinar el esquema lógico y conceptual de la BBDD.

La Figura 4.7 muestra este diagrama donde se diferencian cada uno de los mecanismos de persistencia definidos[2] junto con las entidades que lo conforman. Además, cada entidad puede poseer un estereotipo[3], unos atributos y tipo junto con la especificación en caso de existir del atributo que actuará como clave primaria o PK (*Public Key*) para identificar un registro único en el mecanismo de persistencia. Como particularidad del diagrama, indicar que la entidad `AlertDetails` es una entidad débil lo que quiere decir que su existencia depende de otra entidad (`Alert`) y no posee atributos únicos o clave primaria. Su estereotipo `«concept»` -proveniente de la tecnología Hyperledger Composer (HC)- viene a significar esto mismo.

Figura 4.7: Diagrama Entidad-Relación.



[2]En el caso del servicio de almacenamiento en la nube, al ser un servicio externo no se pueden controlar los atributos que almacena ya que únicamente se envía el fichero. Sin embargo, para hacer entendible el modelo y la información usada en el proyecto se especifican dos atributos y su estereotipo.

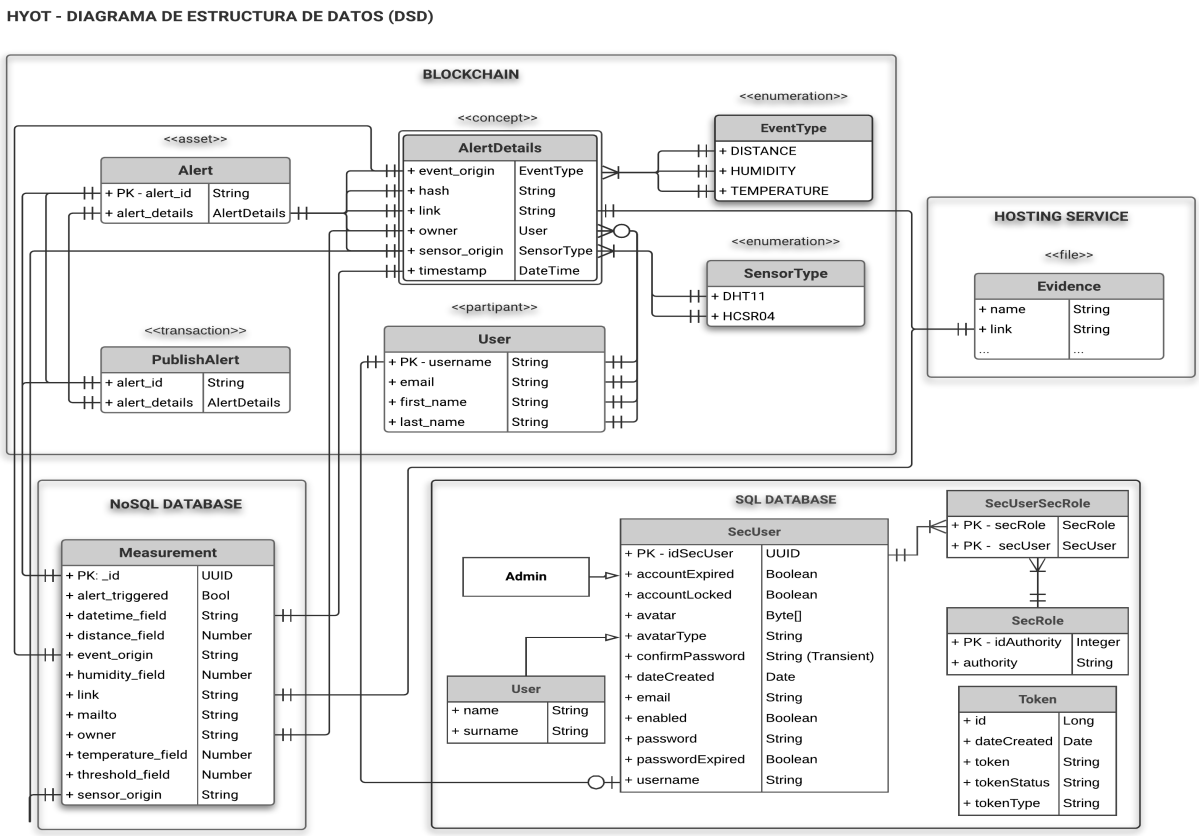
[3]Las entidades que conforman el mecanismo de persistencia BC se explican con más detalle en la sección 5.4.

Diagrama de Estructura de Datos

El diagrama de Estructura de Datos es un modelo conceptual del sistema similar al diagrama Entidad-Relación con la salvedad de que se centra en las relaciones de los elementos dentro de las entidades, en lugar de las relaciones entre las propias entidades.

La Figura 4.8 muestra este diagrama donde al igual que en el diagrama Entidad-Relación las entidades están categorizadas según el mecanismo de persistencia al que pertenecen. En este diagrama la particularidad, como su definición indica, se observa en las relaciones donde son los propios atributos los que las marcan. El significado que se ha querido proporcionar a este diagrama en Hyot, es mostrar qué campos están definidos en cada entidad y cuáles de ellos son compartidos entre diferentes entidades aunque la entidad en sí no esté relacionada, como es el caso de la entidad débil `AlertDetails` y la entidad `Measurement` donde no existe una relación explícita en el modelo pero ambas comparten dos atributos que para la misma instancia poseen el mismo valor.

Figura 4.8: Diagrama de Estructura de Datos.



4.2. Diagramas de secuencia de diseño

En esta sección se muestran los diagramas de secuencia de diseño los cuales representan las interacciones de las entidades del sistema con los casos de uso desde el punto de vista de diseño, conteniendo por tanto detalles acerca de la implementación final de la solución.

- Diagrama de secuencia de diseño: Configurar dispositivo RPi (Figura 4.9 - Parte 1 y Figura 4.10 - Parte 2).

Figura 4.9: Diagrama de secuencia de diseño: Configurar dispositivo RPi - Parte 1.

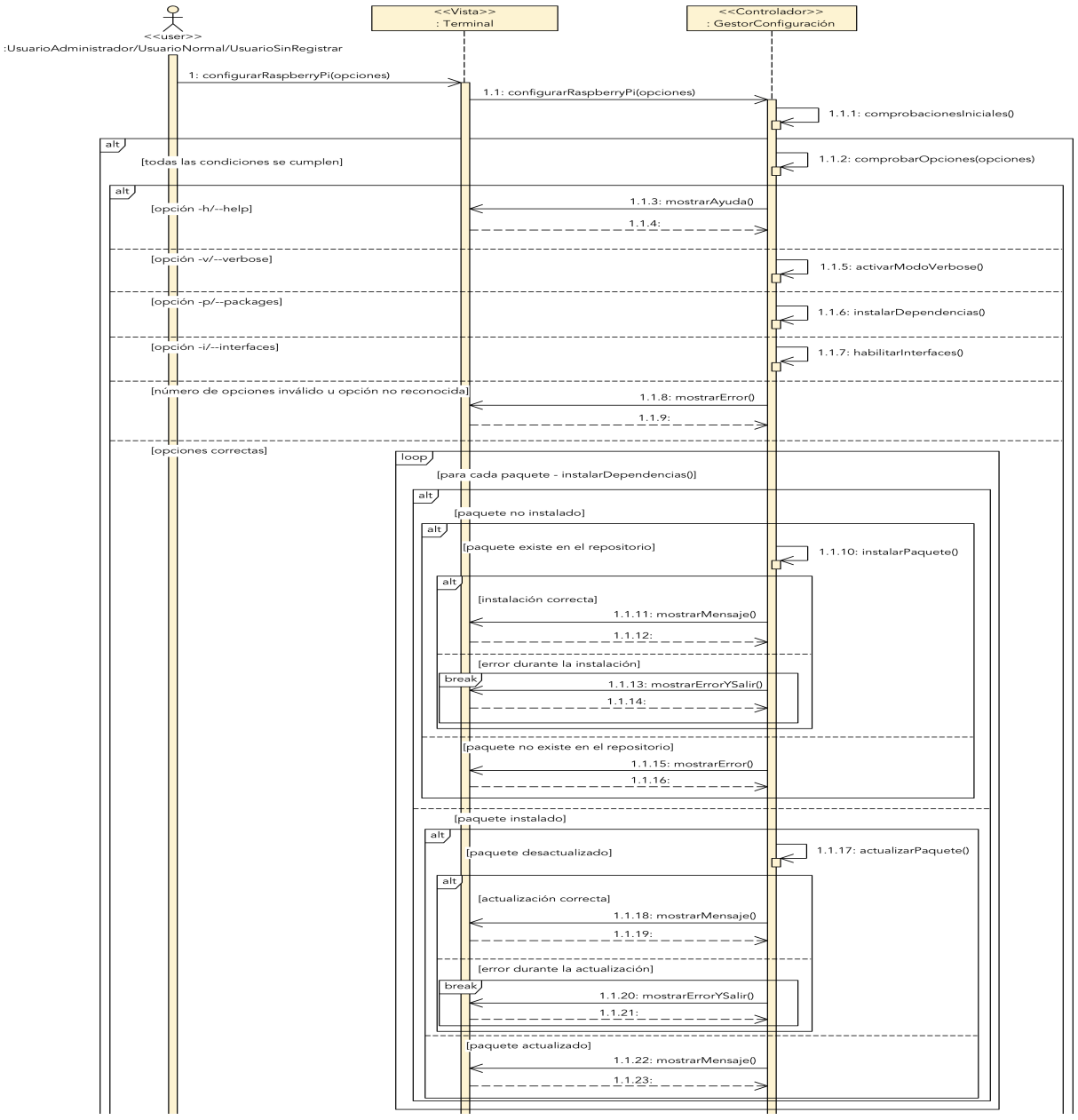
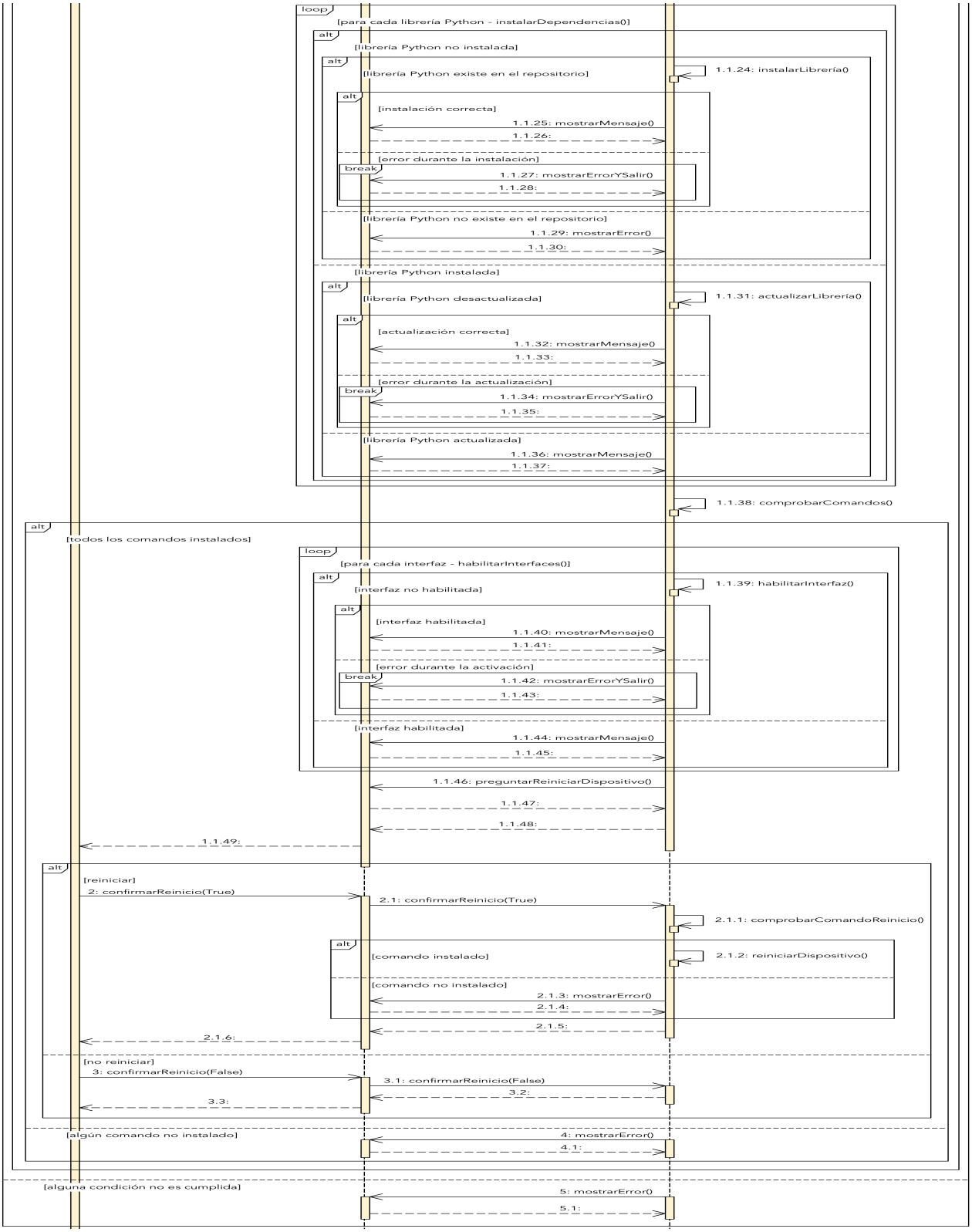
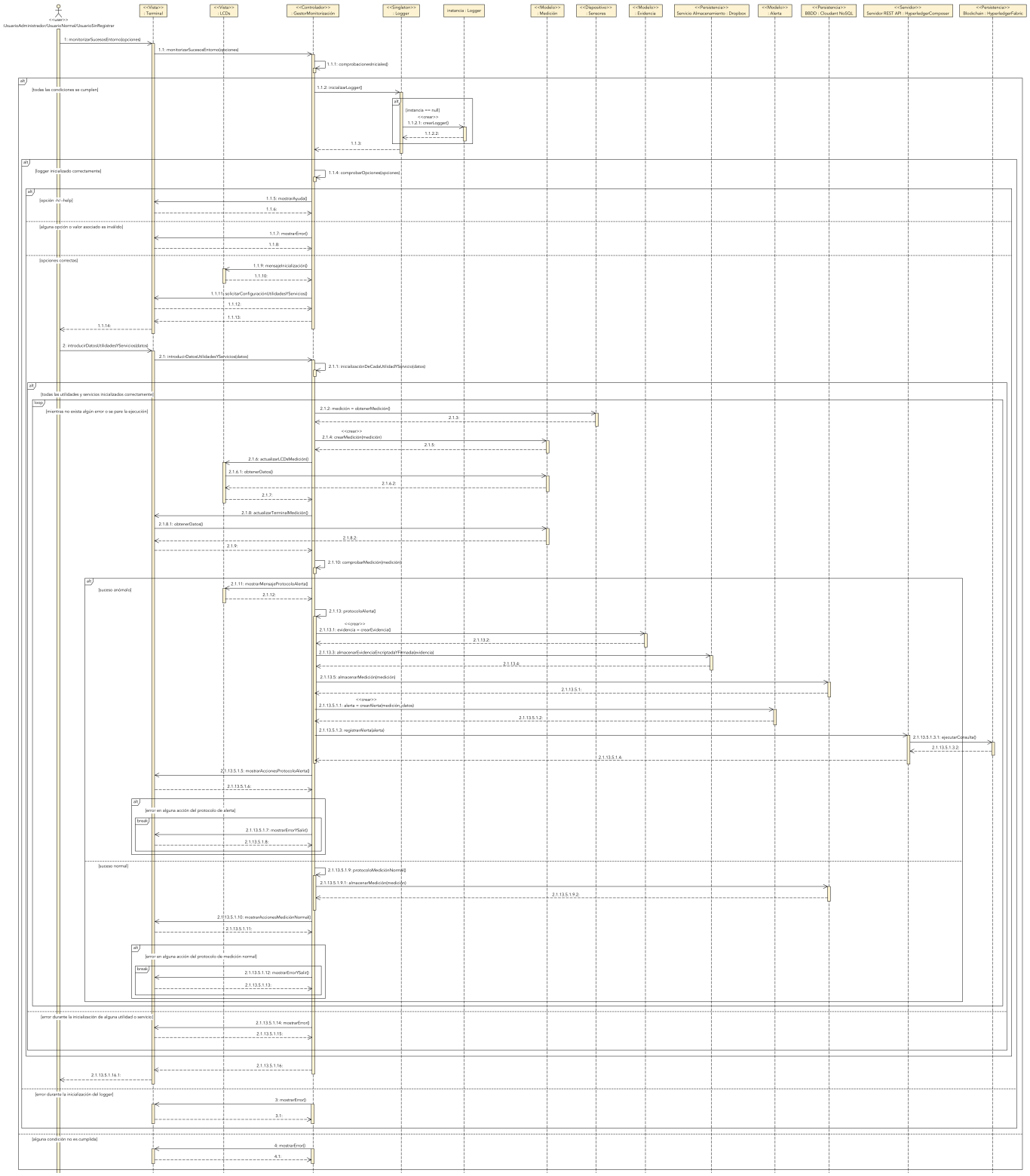


Figura 4.10: Diagrama de secuencia de diseño: Configurar dispositivo RPi - Parte 2.



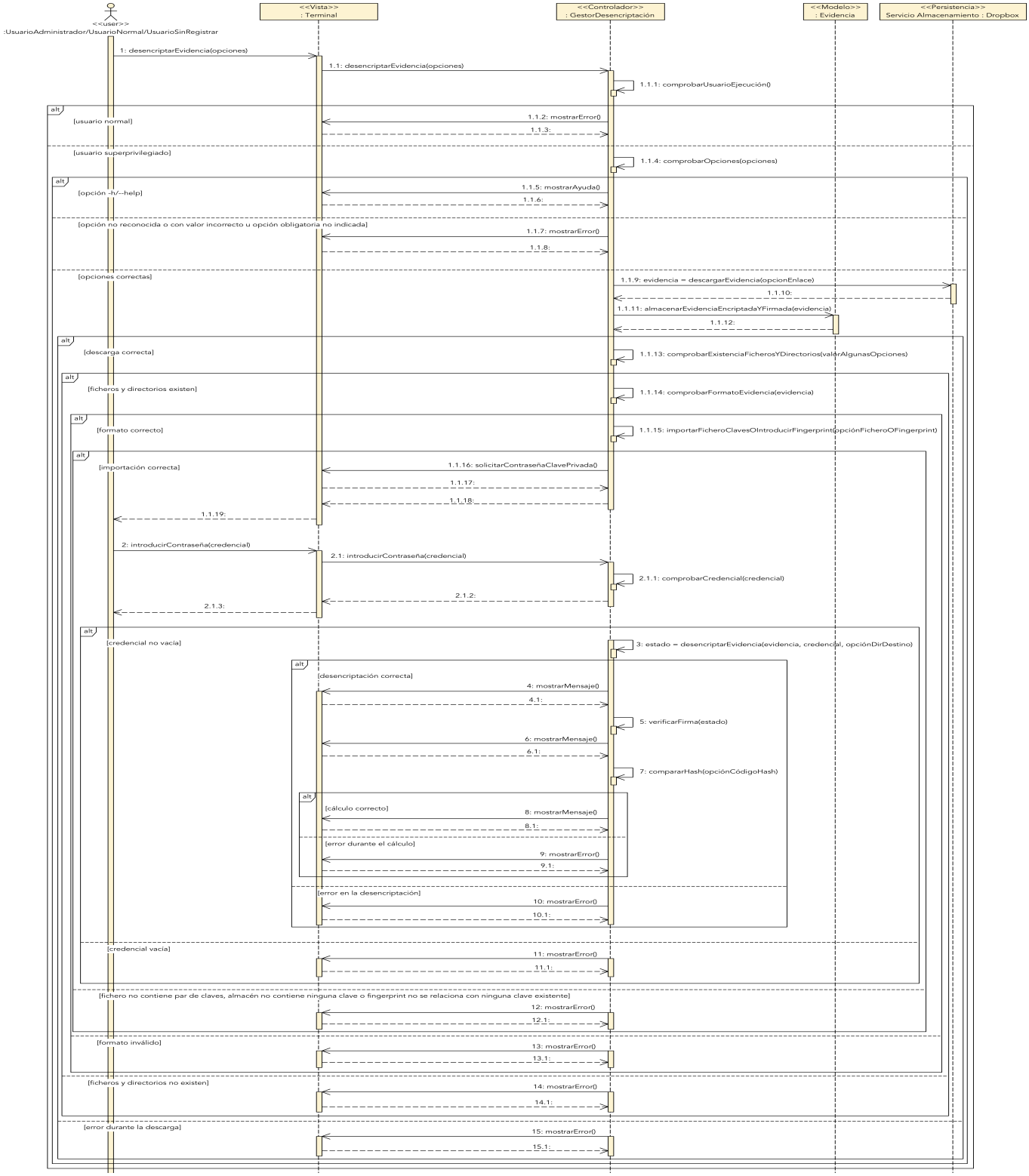
- Diagrama de secuencia de diseño: Monitorizar sucesos del entorno (Figura 4.11).

Figura 4.11: Diagrama de secuencia de diseño: Monitorizar sucesos del entorno.



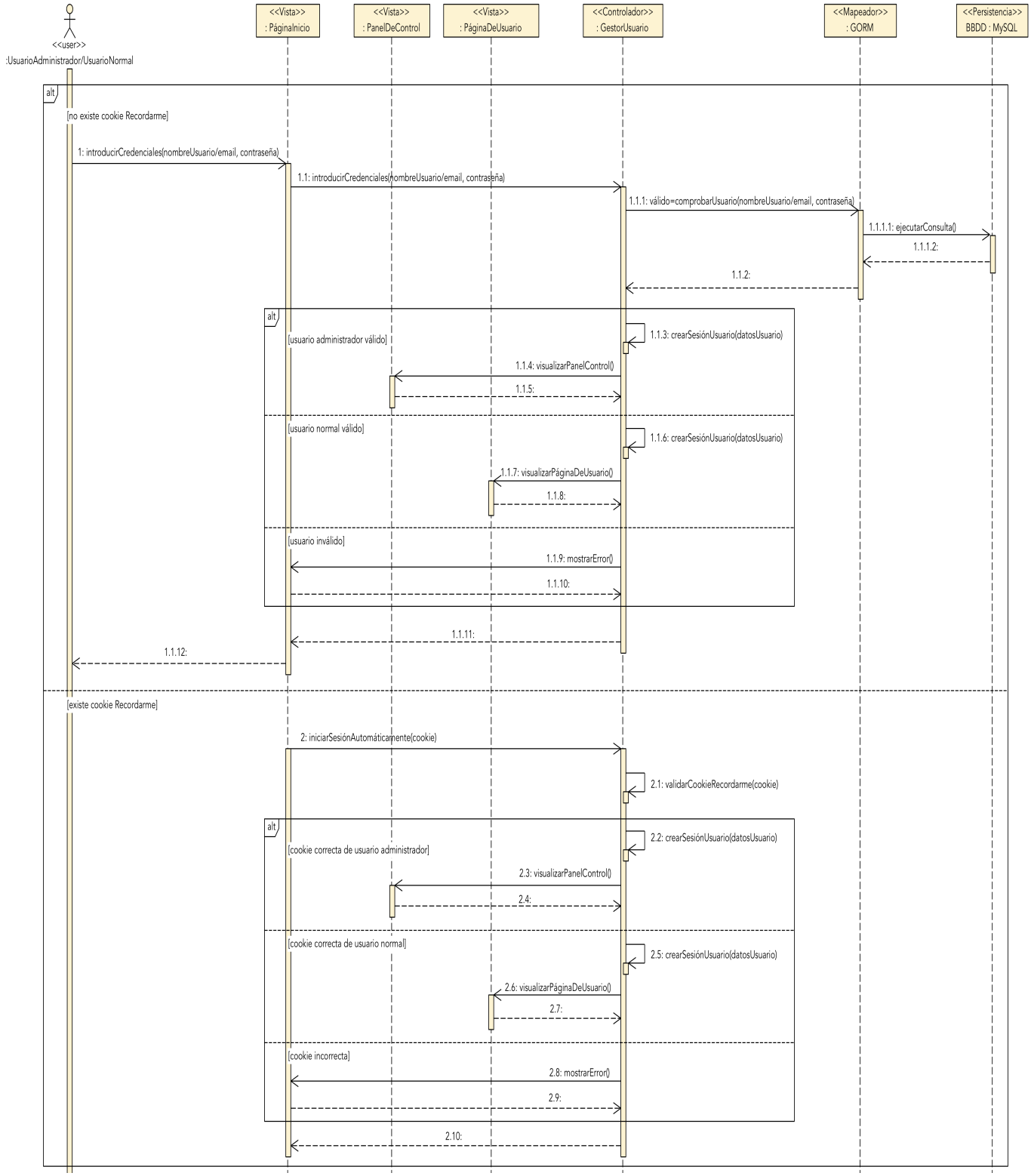
- Diagrama de secuencia de diseño: Desencriptar evidencia (Figura 4.12).

Figura 4.12: Diagrama de secuencia de diseño: Desencriptar evidencia.



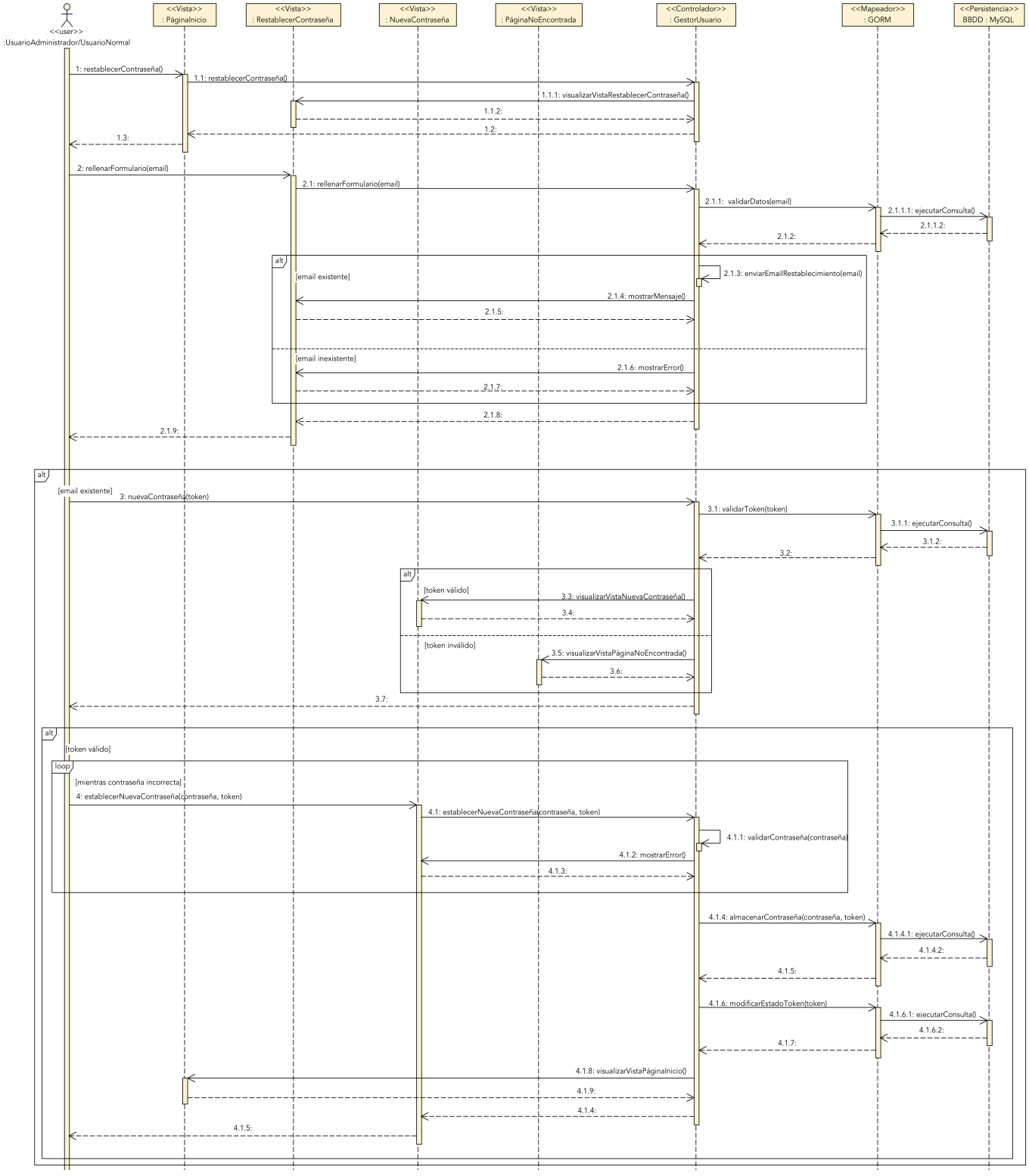
- Diagrama de secuencia de diseño: Iniciar sesión (Figura 4.13).

Figura 4.13: Diagrama de secuencia de diseño: Iniciar sesión.



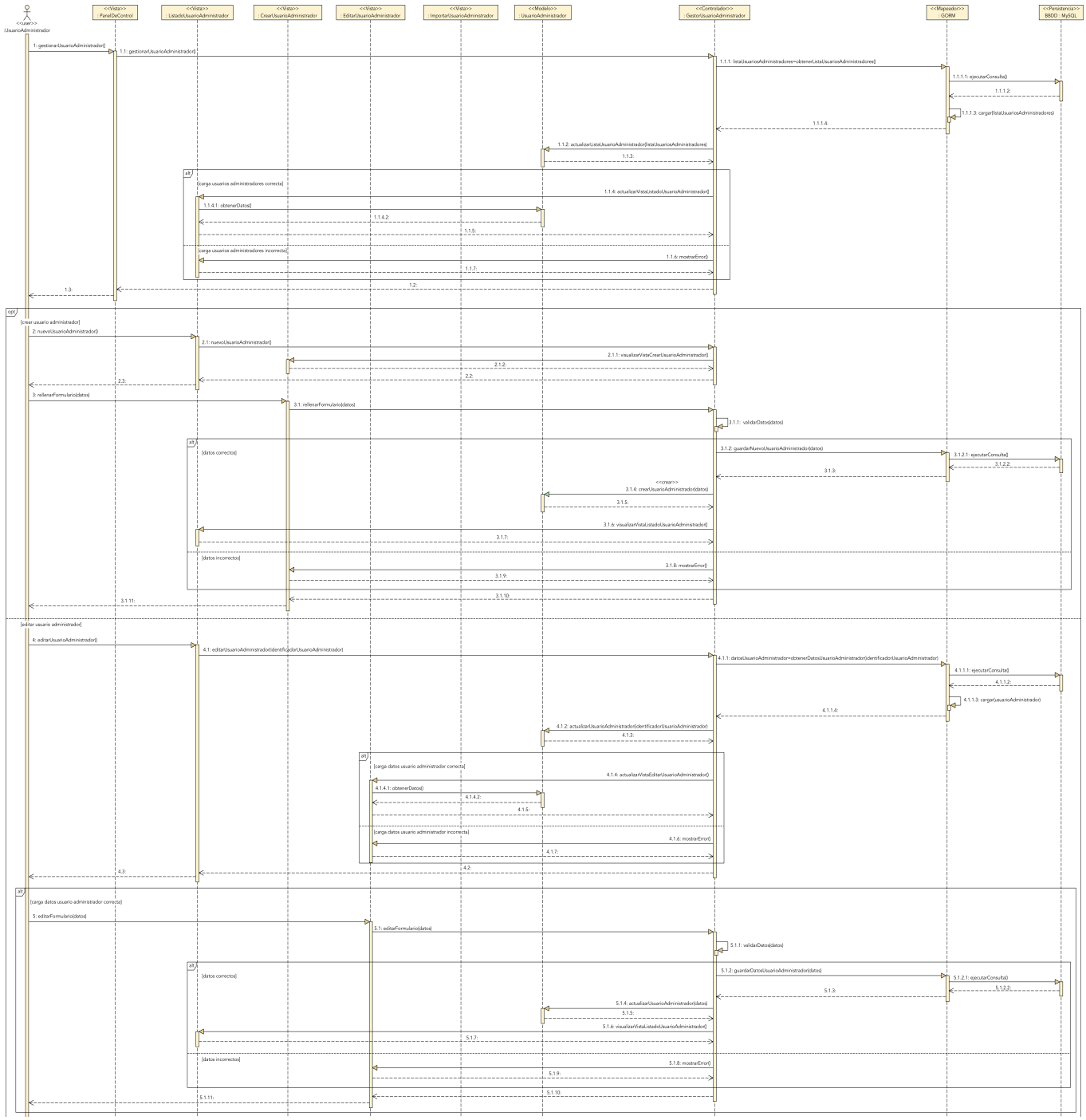
- Diagrama de secuencia de diseño: Restablecer contraseña (Figura 4.14).

Figura 4.14: Diagrama de secuencia de diseño: Restablecer contraseña.



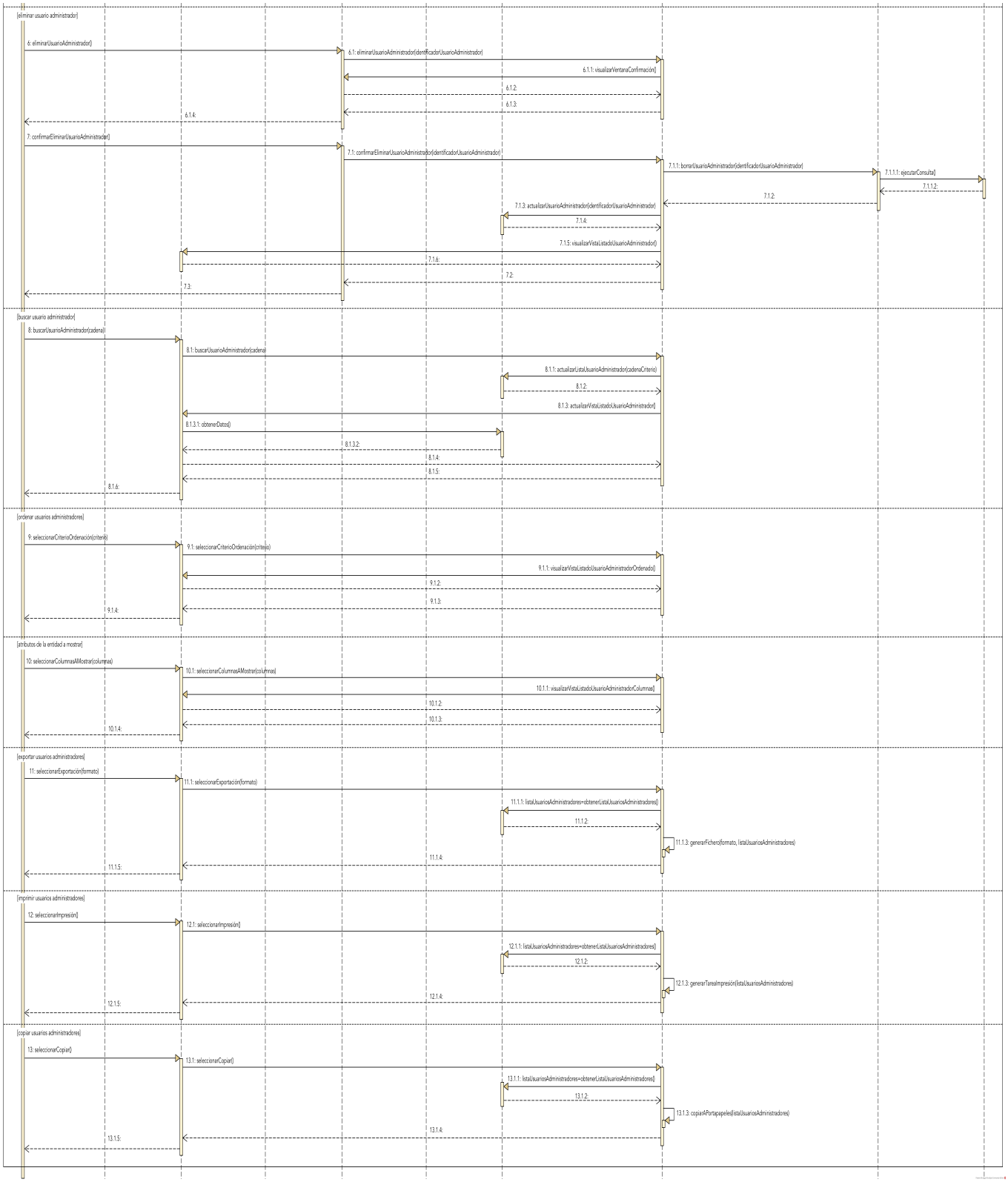
- Diagrama de secuencia de diseño: Gestionar usuario administrador (Figura 4.15 - Parte 1 y Figura 4.16 - Parte 2)[4].

Figura 4.15: Diagrama de secuencia de diseño: Gestionar usuario administrador - Parte 1.



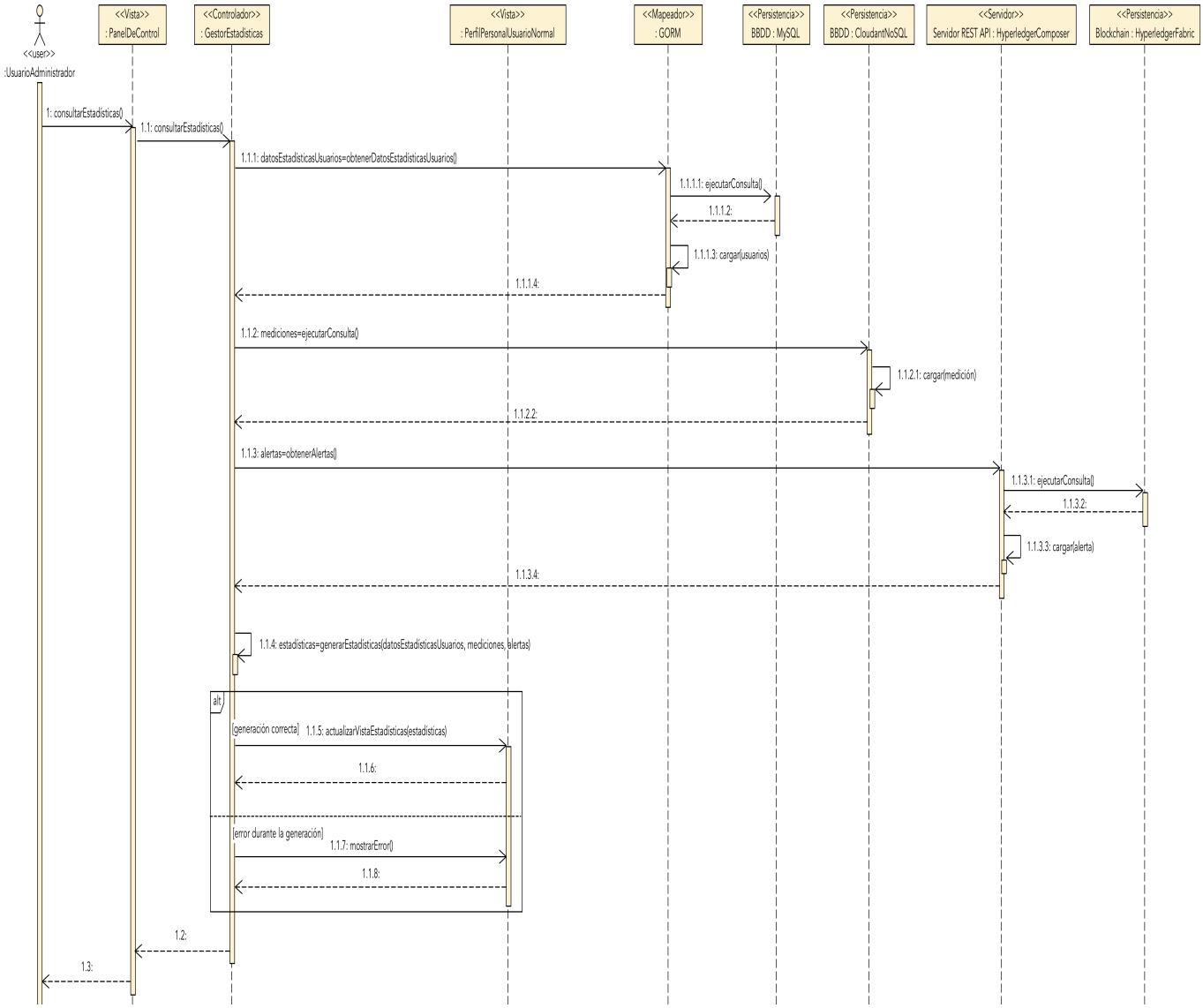
[4]El diagrama de secuencia de diseño para el caso de uso UC-0007: Gestionar usuario normal es similar al presente con la salvedad de adaptar a la entidad correspondiente.

Figura 4.16: Diagrama de secuencia de diseño: Gestionar usuario administrador - Parte 2.



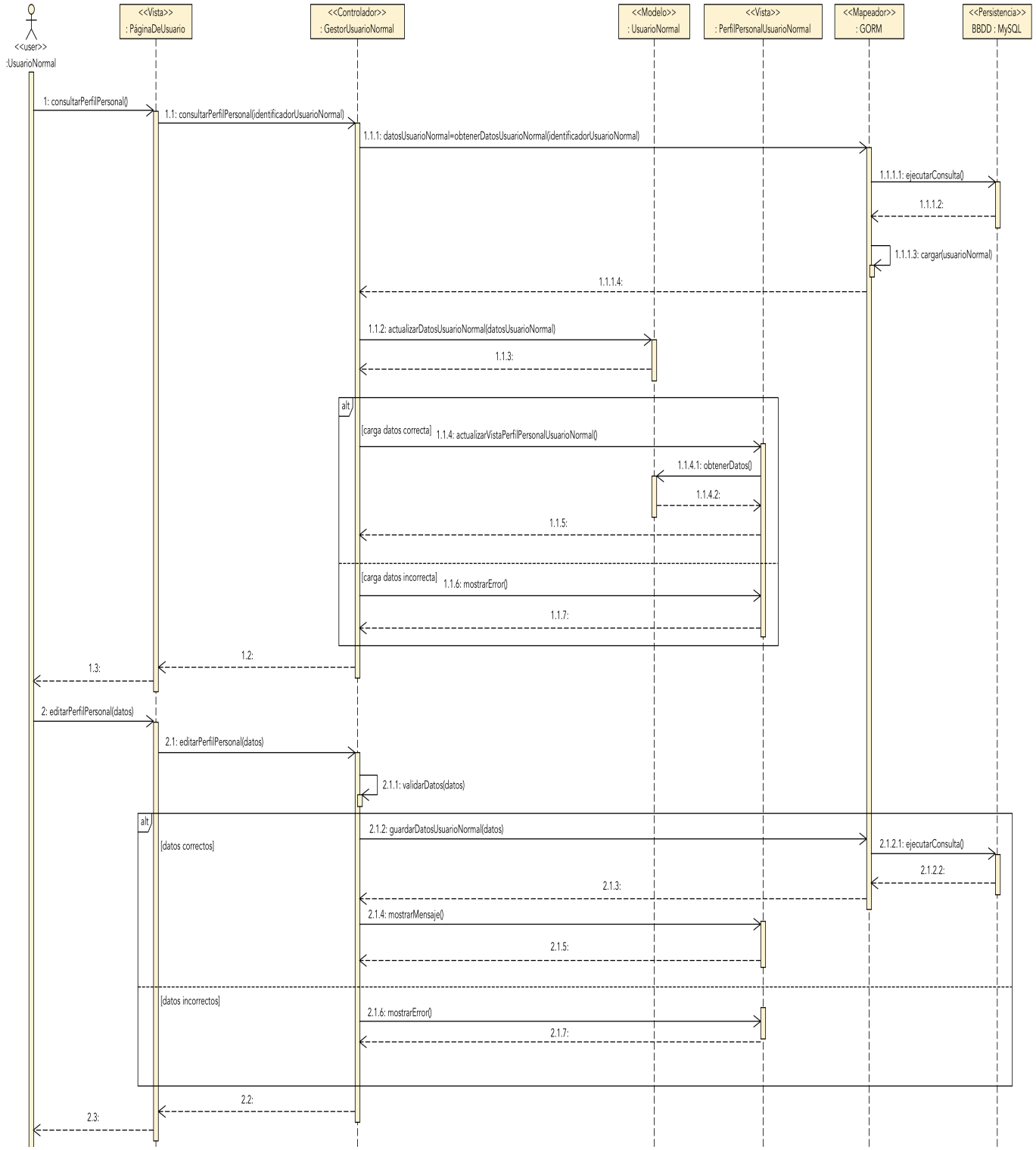
- Diagrama de secuencia de diseño: Consultar estadísticas globales (Figura 4.17).

Figura 4.17: Diagrama de secuencia de diseño: Consultar estadísticas globales.



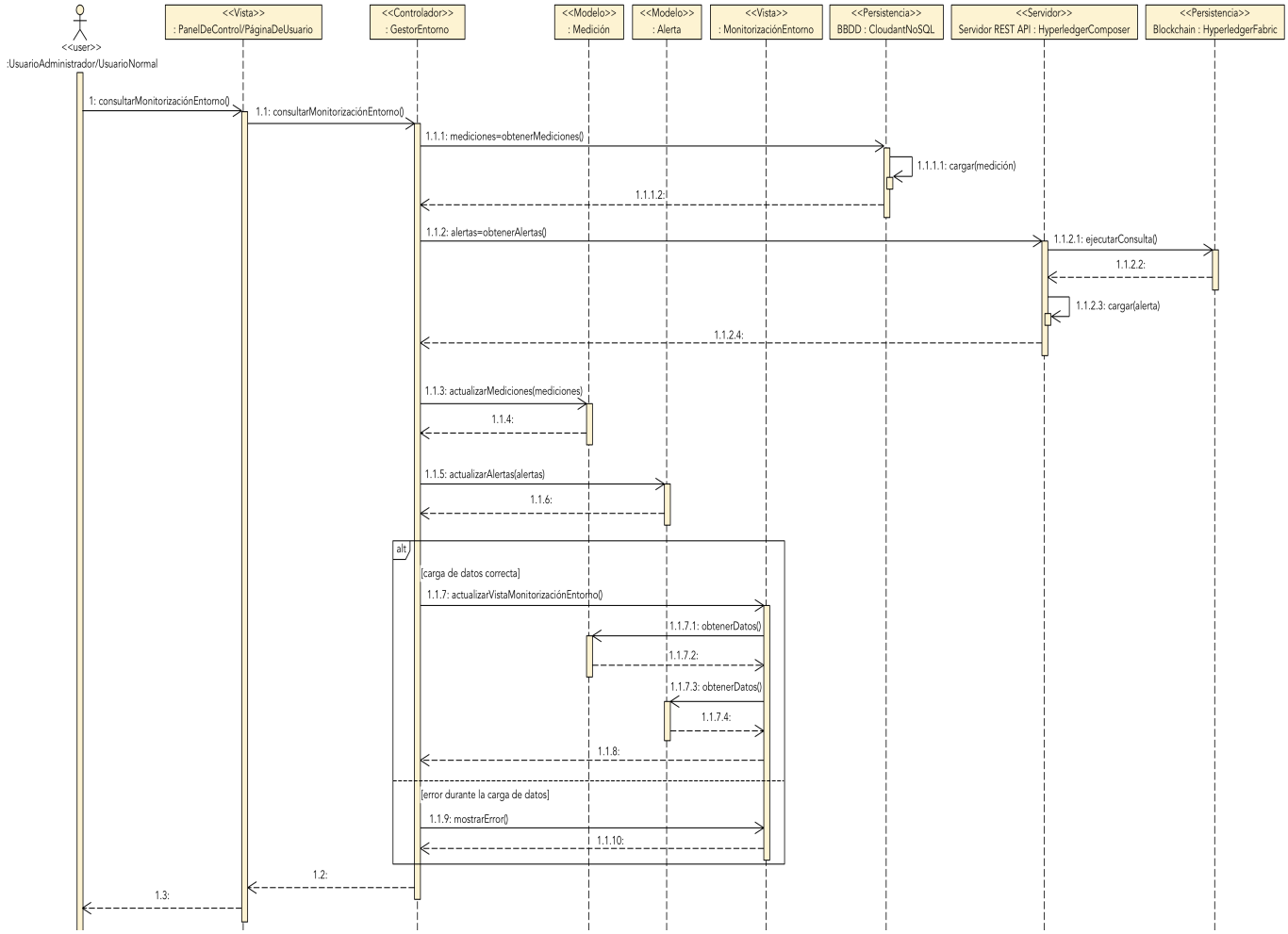
- Diagrama de secuencia de diseño: Consultar o modificar perfil personal (Figura 4.18).

Figura 4.18: Diagrama de secuencia de diseño: Consultar o modificar perfil personal.



- Diagrama de secuencia de diseño: Consumir información sobre la trazabilidad del entorno (Figura 4.19).

Figura 4.19: Diagrama de secuencia de diseño: Consumir información sobre la trazabilidad del entorno.

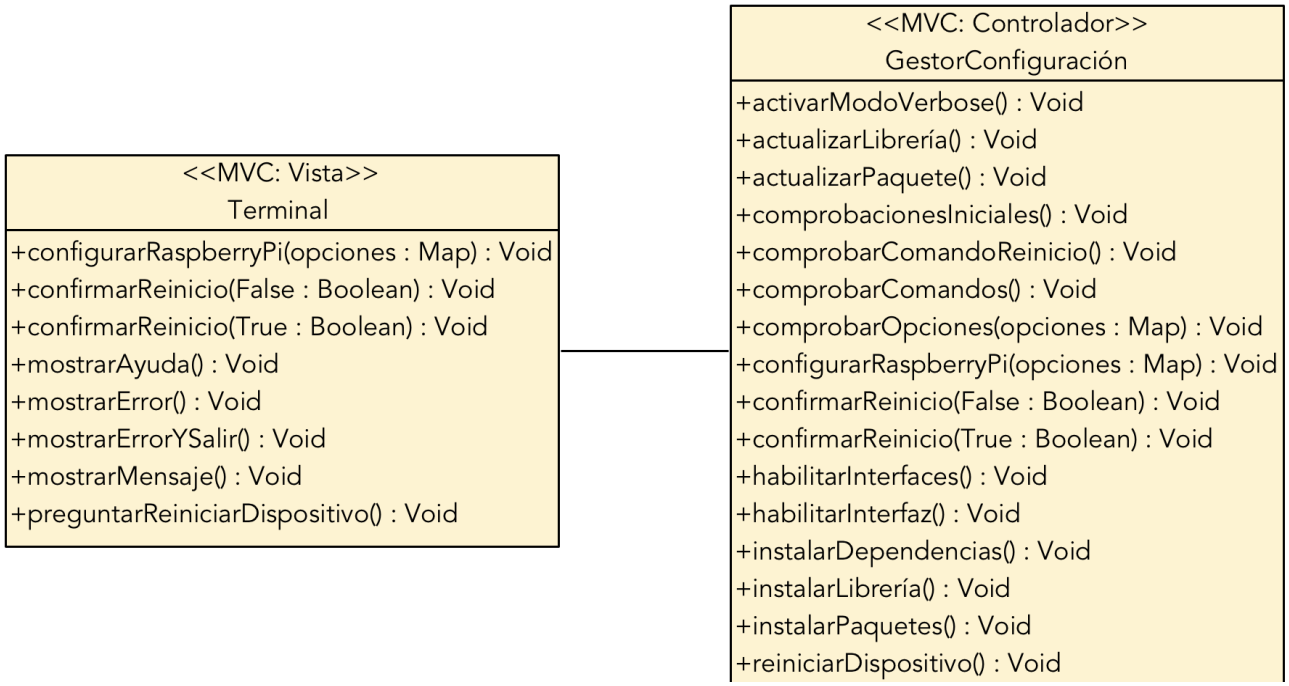


4.3. Diagramas de clases de diseño

En esta sección se muestran los diagramas de clases de diseño los cuales permiten obtener una representación del modelo desde el punto de vista de diseño e implementación de un sistema, es decir, a partir del modelo de dominio obtenido durante la etapa de análisis se incorporan todos aquellos componentes y detalles específicos de diseño con vistas a satisfacer los detalles de la implementación.

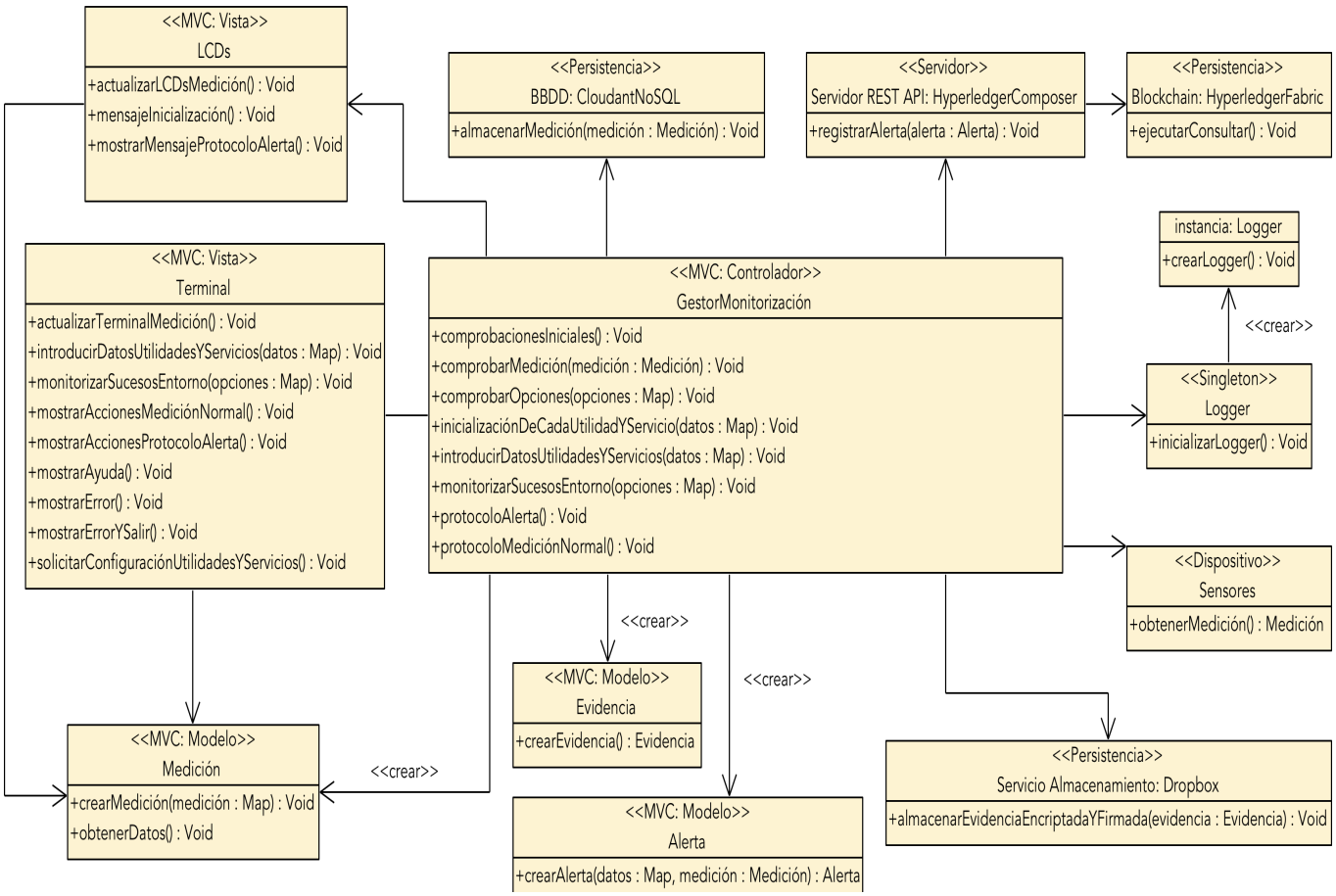
- Diagrama de clases de diseño: Configurar dispositivo RPi (Figura 4.20).

Figura 4.20: Diagrama de clases de diseño: Configurar dispositivo RPi.



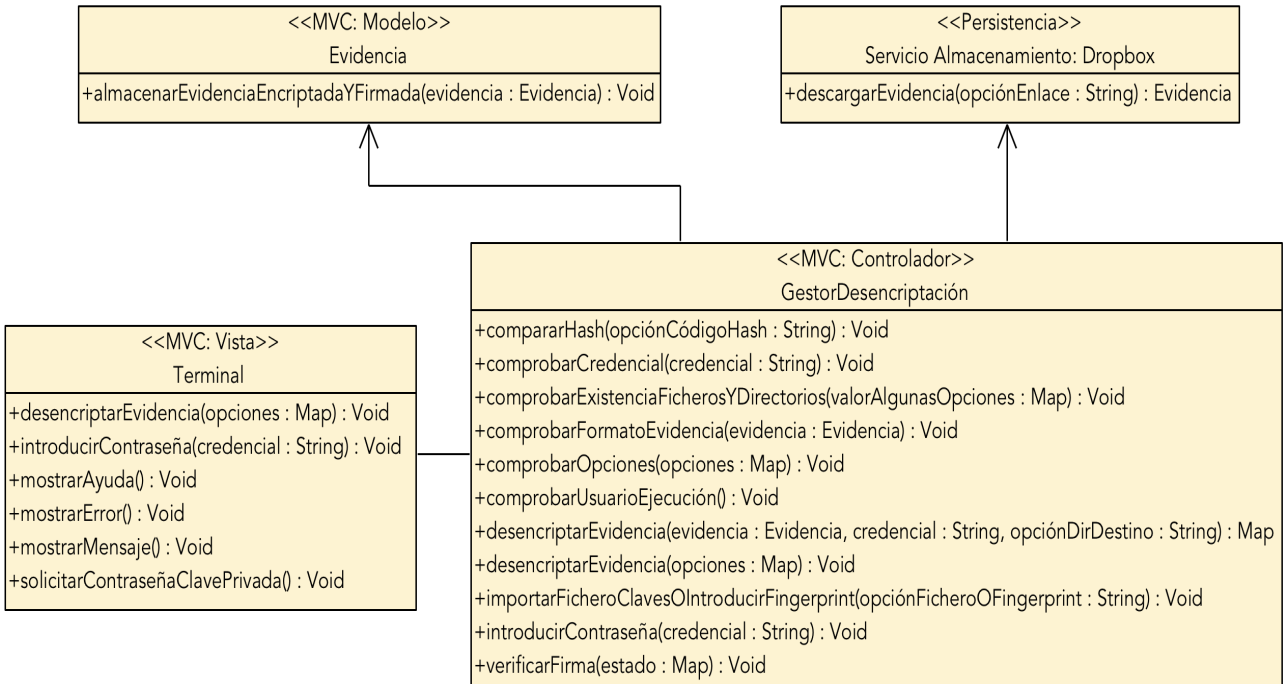
- Diagrama de clases de diseño: Monitorizar sucesos del entorno (Figura 4.21).

Figura 4.21: Diagrama de clases de diseño: Monitorizar sucesos del entorno.



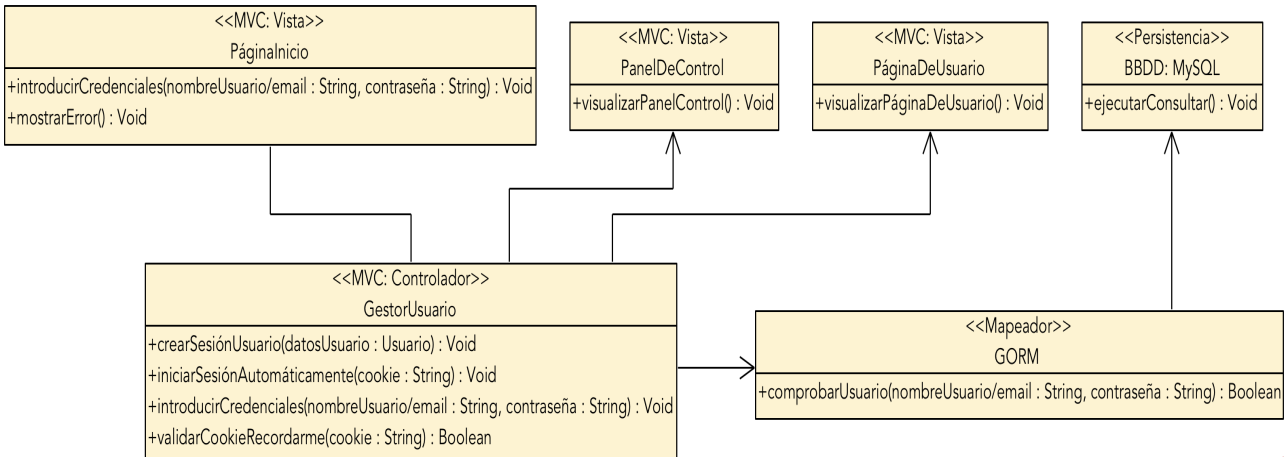
- Diagrama de clases de diseño: Desencriptar evidencia (Figura 4.12).

Figura 4.22: Diagrama de clases de diseño: Desencriptar evidencia.



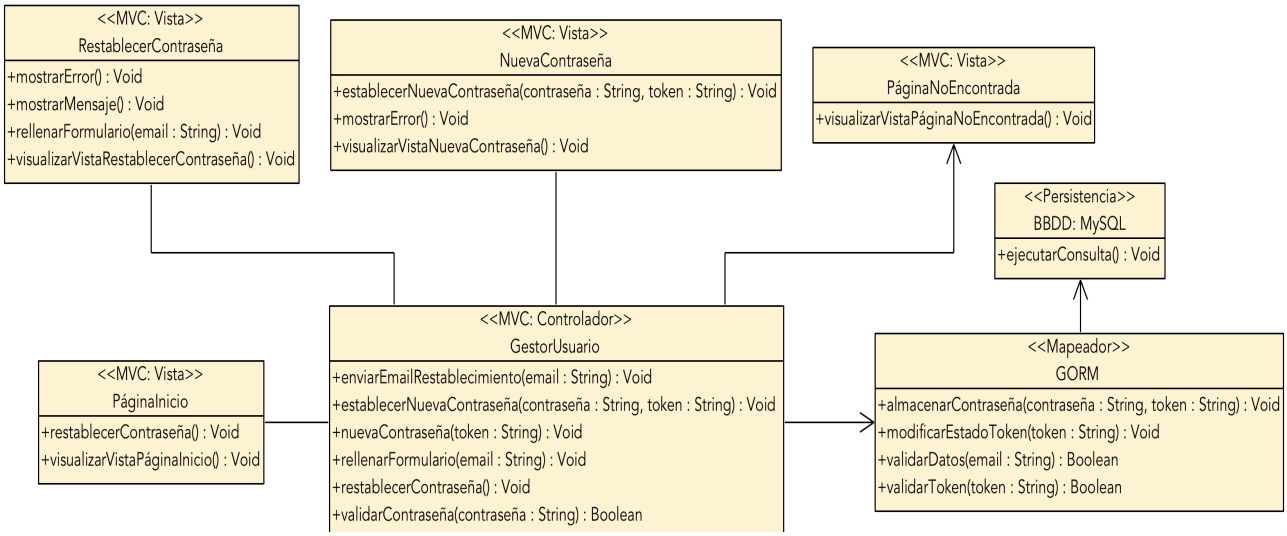
- Diagrama de clases de diseño: Iniciar sesión (Figura 4.23).

Figura 4.23: Diagrama de clases de diseño: Iniciar sesión.



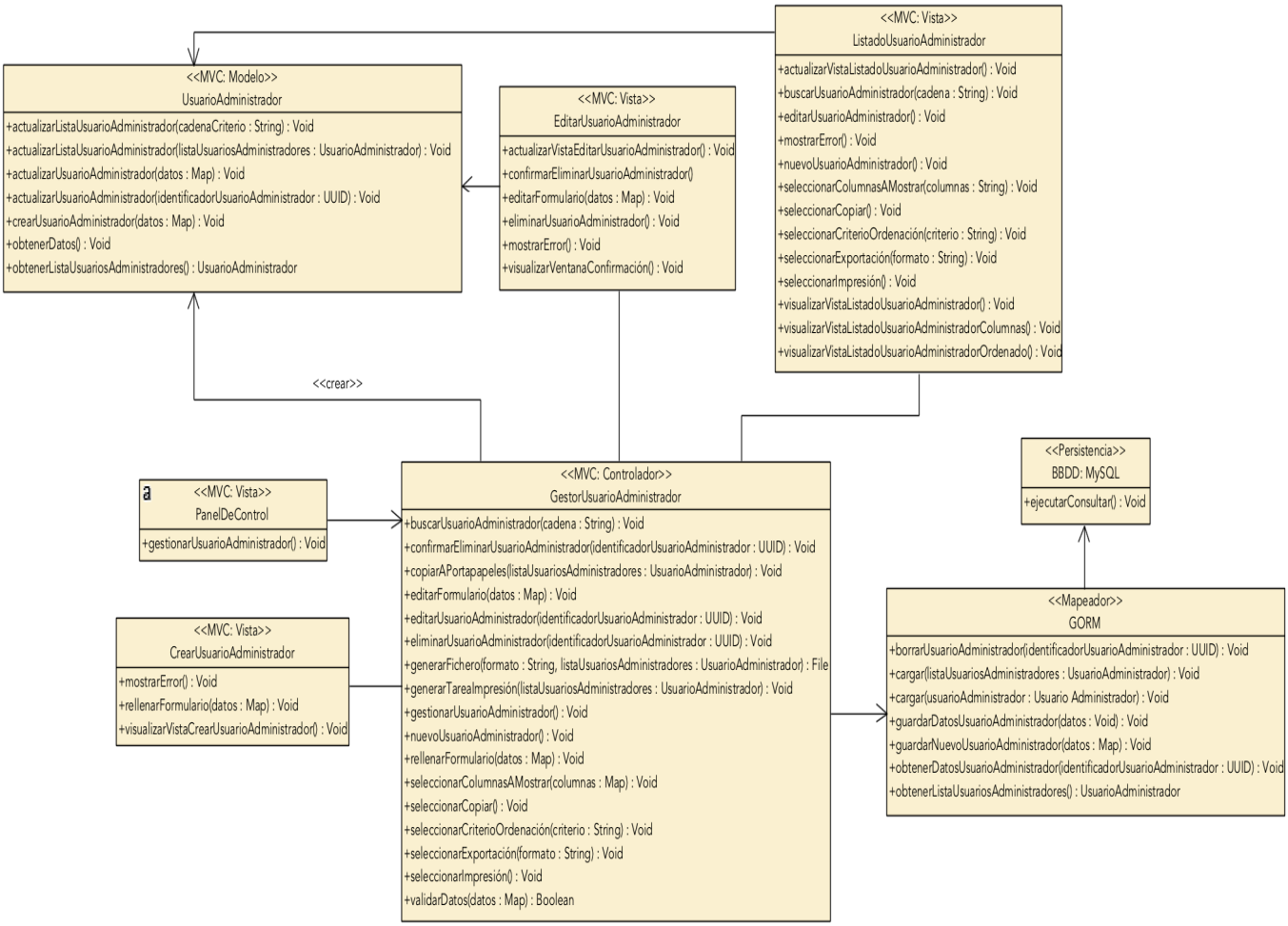
- Diagrama de clases de diseño: Restablecer contraseña (Figura 4.24).

Figura 4.24: Diagrama de clases de diseño: Restablecer contraseña.



- Diagrama de clases de diseño: Gestionar usuario administrador (Figura 4.25)[5].

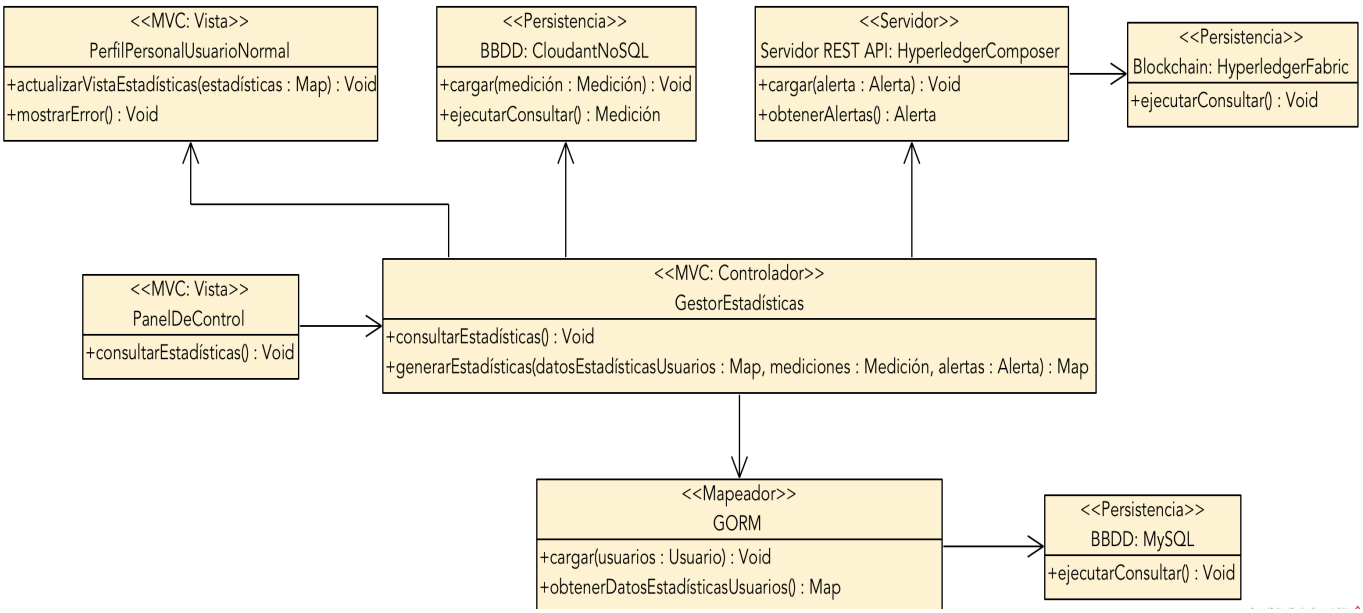
Figura 4.25: Diagrama de clases de diseño: Gestionar usuario administrador.



[5]El diagrama de clases de diseño para el caso de uso UC-0007: Gestionar usuario normal es similar al presente con la salvedad de adaptar a la entidad correspondiente.

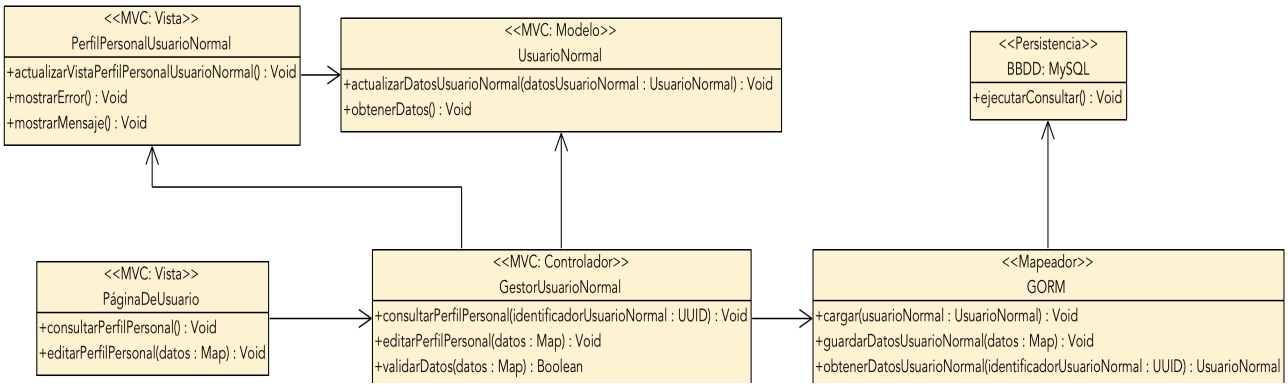
- Diagrama de clases de diseño: Consultar estadísticas globales (Figura 4.26).

Figura 4.26: Diagrama de clases de diseño: Consultar estadísticas globales.



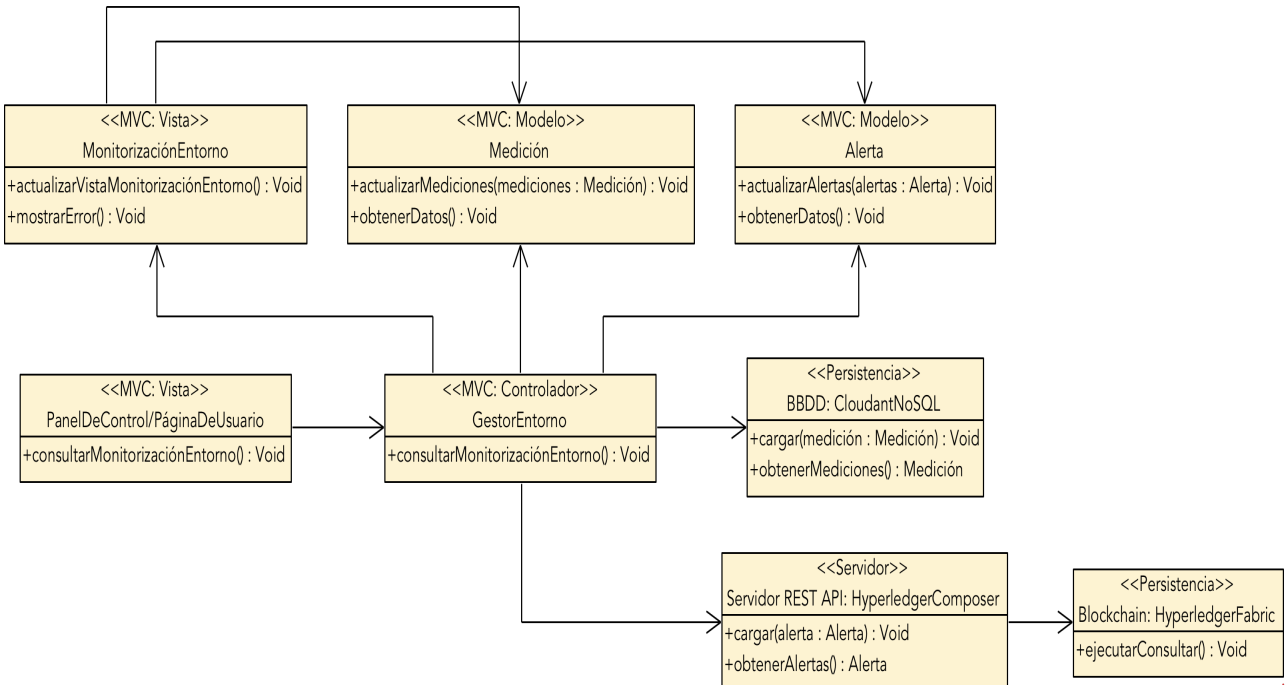
- Diagrama de clases de diseño: Consultar o modificar perfil personal (Figura 4.27).

Figura 4.27: Diagrama de clases de diseño: Consultar o modificar perfil personal.



- Diagrama de clases de diseño: Consumir información sobre la trazabilidad del entorno (Figura 4.28).

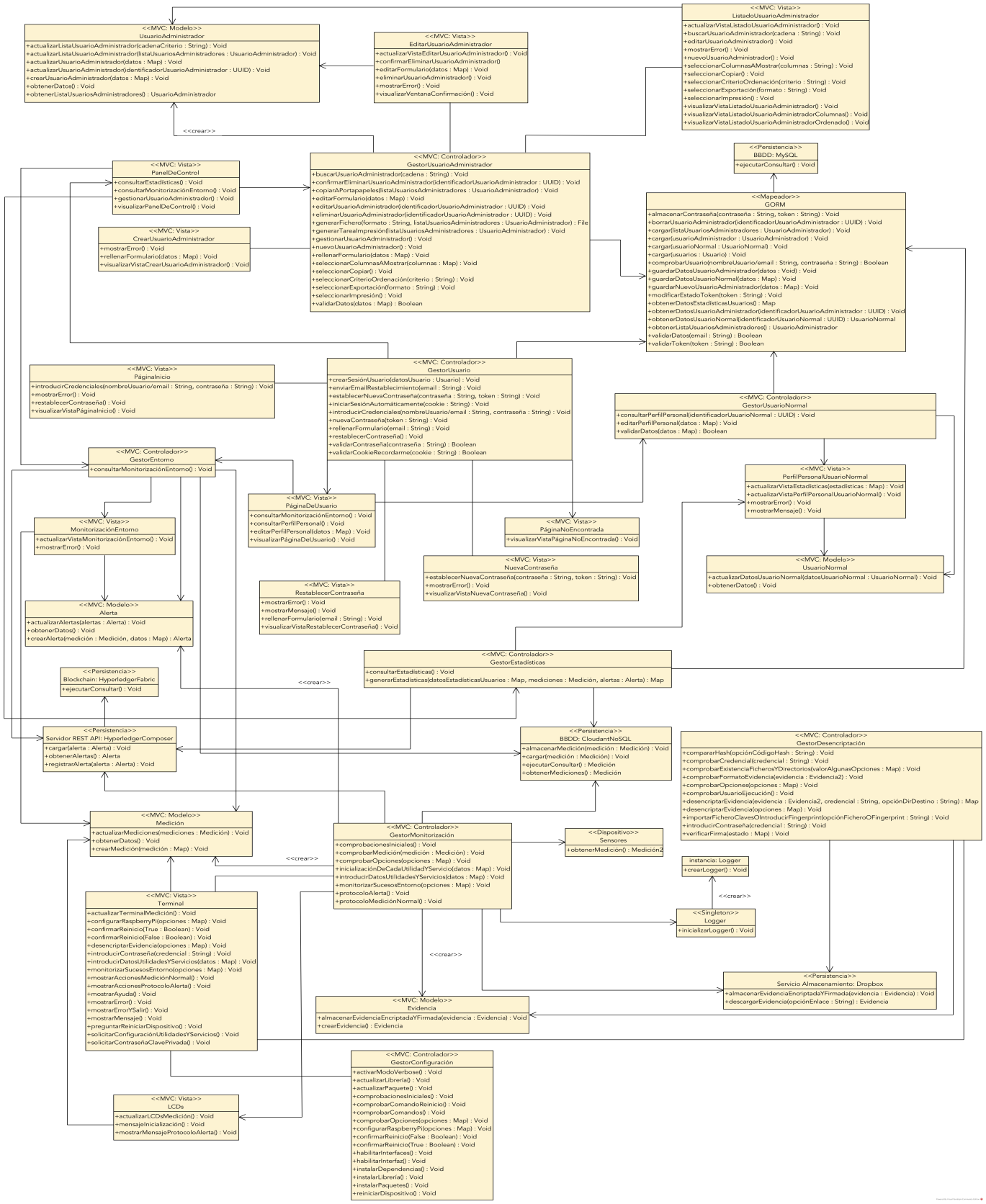
Figura 4.28: Diagrama de clases de diseño: Consumir información sobre la trazabilidad del entorno.



4.4. Diagrama final de clases de diseño

El diagrama final de clases de diseño (Figura 4.29) contiene la representación completa del modelo desde el punto de vista de diseño e implementación de un sistema, es decir, se trata de la fusión de todos los diagramas de clases de diseño expuestos en la sección anterior.

Figura 4.29: Diagrama final de clases de diseño.



Capítulo 5

Implementación

Todo sistema *software* se encuentra respaldado por un trabajo previo que lo conforma. Un aspecto importante es mostrar al lector la información interna más destacada sobre la implementación tecnológica llevada a cabo ya que de esta manera podrá conocer de una forma más concreta los entresijos del sistema, la funcionalidad, etc.

5.1. Hyot

Hyot es la prueba de concepto (*Proof of Concept* -PoC-) de código abierto[1] desarrollada para llevar a cabo la trazabilidad de un entorno controlado de IoT (*Internet of Things*) mediante la tecnología Hyperledger Fabric (HF). Si bien es cierto que esta trazabilidad se puede realizar utilizando metodologías tradicionales, como son bases de datos o ficheros de *logs* para almacenar la información, también es evidente que se carece de una falta de seguridad y garantía de que esta información ha perdurado inmutable desde su registro.

Aprovechando el respaldo de todas las ventajas que proporciona el concepto de Blockchain (BC), Hyot mantiene parte de su persistencia en una BC de carácter permisionada lo cual incrementa favorablemente la seguridad frente a las públicas ya que se puede controlar quién puede acceder y con qué privilegios al existir varios niveles de privacidad y de membresía proporcionando así mayor confidencialidad. De esta forma la información no es expuesta públicamente a cualquier usuario ajeno a la misma y siempre está identificada al participante que la registra por lo que favorece la trazabilidad de los datos. Pero solamente una parte de toda la información trazada del entorno es registrada en la BC, en concreto aquella que interesa salvaguardar de una posible alteración ya que es la que proporcionará el resguardo y la prueba de que un suceso irregular realmente sucedió en un marco temporal. El suceso registrado a partir del cual se genera una evidencia -vídeo del entorno- se guarda en una tercera parte de confianza (TPC), como puede ser un servicio de almacenamiento en la nube, debido a su tamaño y al formato que presenta. Este

[1] Consulte el anexo G para saber cómo conseguir el código y cómo poner en marcha esta PoC.

agente externo puede no encontrarse 100 % securizado por lo que la evidencia almacenada debe ser protegida mediante el cifrado de su contenido y posterior firmado. La encriptación proporciona al contenido una codificación de tal manera que la información no sea legible de forma directa otorgando así confidencialidad y privacidad. Por su parte, el firmado ayuda a asegurar:

- Integridad de los datos: la evidencia no ha sido alterada de su forma original.
- Mensaje de autenticación o prueba de origen: la evidencia en realidad procede del supuesto remitente.
- No repudio: el emisor no puede negar la autenticidad de la evidencia que envió y firmó.

El resto de información que es meramente informativa y que permite ampliar el contexto de la trazabilidad efectuada es almacenada en una base de datos (BBDD). También, cabe destacar que es fácilmente escalable a ampliar a un mayor rango de cobertura en cuanto a IoT se refiere, indistintamente de usar dispositivos del mismo tipo o variedad en ellos.

Esta solución, que es altamente parametrizable y configurable por el usuario, gestiona de forma transparente éste una serie de sucesos -temperatura, humedad y distancia- del entorno que son monitorizados constantemente con una frecuencia por defecto de 3 segundos y en tiempo real desde fuentes de entradas de datos, como son los sensores, conectados a un ordenador de placa reducida (*Single Board Computer* -SBC-) como es la Raspberry Pi (RPi), siendo la última versión disponible (versión 3) la utilizada en el proyecto. Esta información recogida es analizada para determinar si en un instante concreto de tiempo los valores leídos se consideran anómalos o no y por tanto decretar si en el entorno se está produciendo un acontecimiento desautorizado. En todo momento, las acciones ejecutadas y las mediciones efectuadas se notifican al usuario tanto a través del terminal como a través de los dispositivos de salida con los que cuenta el prototipo *hardware*[2], como son los LCDs (*Liquid Crystal Display*) y el LED (*Light-Emitting Diode*), este último activándose solamente en caso de accionar el protocolo de alerta.

En el caso de que la medición –monitorización del entorno en un instante de tiempo- actual no reporte ningún caso anómalo, se procede con el protocolo de medición normal compuesto por la siguiente acción:

1. Almacenar la medición en la BBDD para que quede constancia de que en ese momento temporal el entorno controlado no presenta ninguna incidencia. Para cada medición se almacenan los siguientes campos:
    - Identificador de la medición.

[2]Consulte los anexos D, E y F para obtener más información sobre la implementación *hardware* de Hyot.

- Marca temporal (*timestamp*) en la que ocurre el suceso.
- Valor del evento temperatura en la medición actual.
- Valor del evento humedad en la medición actual.
- Valor del evento distancia en la medición actual.
- Indicación de si se produce un suceso anómalo y por tanto se activa o no el protocolo de alerta.
- Sensor que origina el protocolo de alerta.
- Evento del sensor que origina el protocolo de alerta.
- Umbral límite del evento que lanza el protocolo de alerta.
- Enlace al servicio de almacenamiento en la nube donde se ubica la evidencia encriptada y firmada.
- Dirección *email* del destinatario para efectos de notificación.
- Usuario que registra la alerta en la BC en caso de que el suceso sea anómalo.

Por contra, si la lectura actual de valores presenta datos anómalos que superan unos umbrales preestablecidos considerados como posible indicio de una incidencia en el entorno, el procedimiento a seguir es más exhaustivo debido a que esta situación es la se quiere certificar su originalidad. Una incidencia no es sino una acción no controlada que tiene lugar en el entorno que se está vigilando, y origina la ejecución de un protocolo de alerta que comprende:

1. Captura de un vídeo -por defecto 10 segundos de duración- a través de una cámara (dispositivo Picamera) conectada a la RPi que representará la evidencia.

2. Cálculo del valor *hash* del contenido original de la evidencia, aplicando la función hash SHA-3 (anteriormente conocido como *keccak*), que actúa como prueba de integridad y por tanto se almacena únicamente en la BC.

3. Encriptación y firmado de la evidencia con la utilidad GPG puesto que se ha asumido un modelo de confianza nula en el cual toda la información debe ser protegida.

4. Almacenamiento en la nube de la evidencia. Este tipo de servicios son considerados TPC que pueden poner en riesgo la privacidad de las evidencias por lo que es necesario el paso anterior de cifrado y firmado.

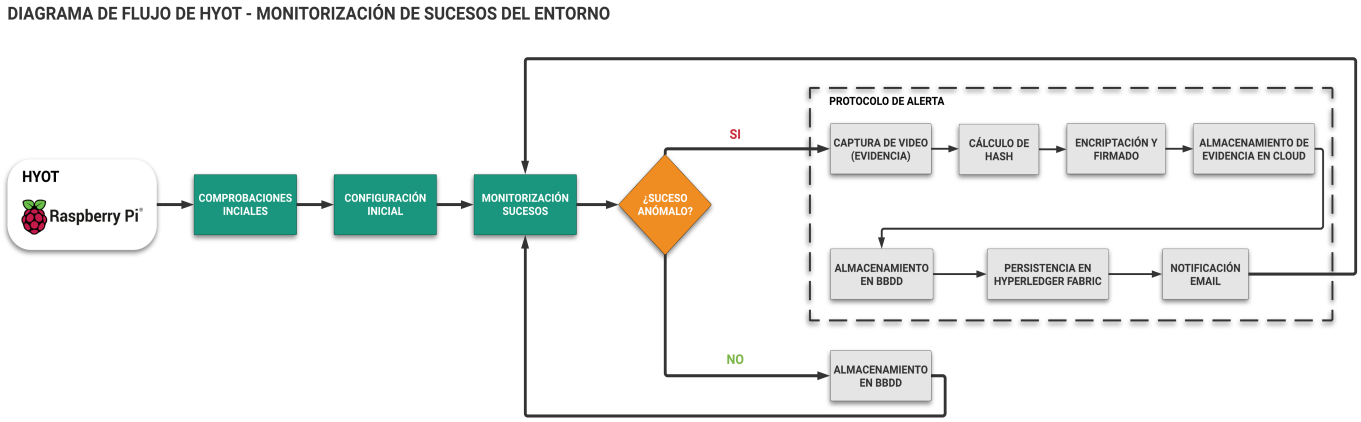
5. Volcado de la información completa de la medición en la BBDD para que quede constancia de que en ese momento temporal el entorno vigilado presenta una lectura no controlada. Este método de almacenamiento como no garantiza la no alteración, la propiedad (código *hash*) que permite garantizar la originalidad del hecho no es almacenada.

6. Registro de la incidencia en HF. El punto central de Hyot es garantizar que el registro de un suceso anómalo no ha sido indebidamente modificado, de forma que una vez registrada la alerta se tenga total certeza respecto a su integridad y veracidad. Este objetivo se consigue con esta tecnología ya que proporciona un protocolo de consenso distribuido para la protección de integridad y un control de acceso que permite identificar a los agentes que introducen datos en la BC. Por ese motivo, en este método de persistencia se almacena de forma exclusiva el valor *hash*.

7. Notificación del suceso no controlado mediante el envío de un *email* a la dirección de correo electrónico destino configurada. Esta comunicación es importante para que el usuario pueda conocer el estado del entorno en todo momento lo antes posible y ver que está sucediendo y poder tomar decisiones en caso necesario.

La Figura 5.1 muestra el mapa de flujo de forma superficial de la monitorización de sucesos del entorno donde durante su ejecución, indistintamente de la activación o no del protocolo de alerta, se gestiona exhaustivamente cualquier error que se pueda producir, finalizando la ejecución de manera instantánea y notificando al usuario por la interfaz de la terminal. Además, si el error se produce durante la etapa de monitorización de sucesos, el usuario tiene la certeza de que si algo no funciona correctamente se le notifica instantáneamente mediante el envío de un *email* adjuntando el *log* con los pasos ejecutados hasta el momento -si dicha funcionalidad está habilitada-.

Figura 5.1: Diagrama de flujo - Monitorización de sucesos del entorno.



A modo resumen, se puede detallar que Hyot se compone de tres componentes principales[3]:

- Componente de monitorización de sucesos del entorno a través de sensores situados en una RPi.

[3]Consulte el anexo H para conocer la funcionalidad de cada componente.

- Protocolo de registro de incidencias en la BC de HF, y almacenamiento de evidencias en la nube.
- Sistema web que actúa como cliente y consume en tiempo real la información registrada por la RPi.

A mayores de los componentes principales, dos componentes adicionales son ofrecidos con el fin de completar el proyecto y facilitar las tareas de:

- Configuración inicial del dispositivo RPi para la posterior ejecución del componente de monitorización de sucesos del entorno. La Figura 5.2 muestra el mapa de flujo de forma superficial de este proceso donde se han omitido acciones y/o comprobaciones de menor importancia y la gestión de errores por simplicidad, en cuyo caso de reproducción se finaliza la ejecución de manera instantánea notificando al usuario por la interfaz del terminal.
- Desencriptación de evidencia previamente encriptada y firmada con GPG. En este proceso, además del proceso de desencriptación, se muestra la firma y se verifica la integridad del contenido de la evidencia obtenida a través de la comparación del valor *hash* calculado sobre el contenido de la evidencia tras su descifrado y el valor *hash* que inicialmente se había almacenado en la BC cuando se produjo la incidencia. En este punto de comprobación reside la confianza que se deposita en esta tecnología ya que en caso de diferir los valores, se tiene la certeza de que la evidencia almacenada en la nube fue alterada.

Figura 5.2: Diagrama de flujo - Configuración inicial de la RPi.

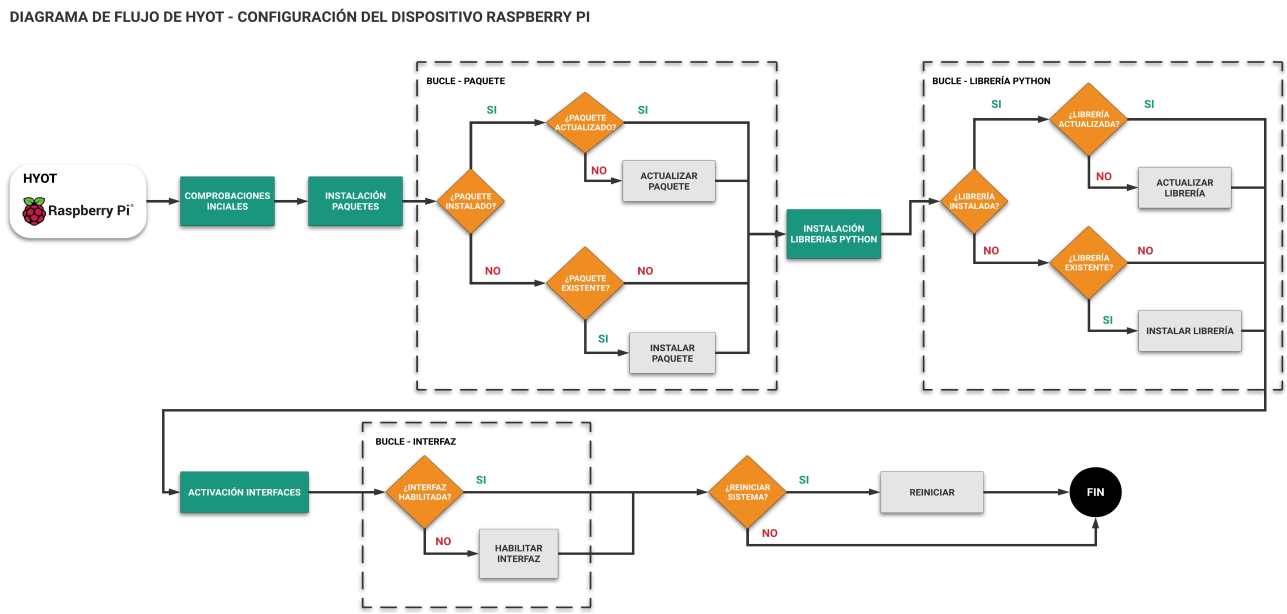
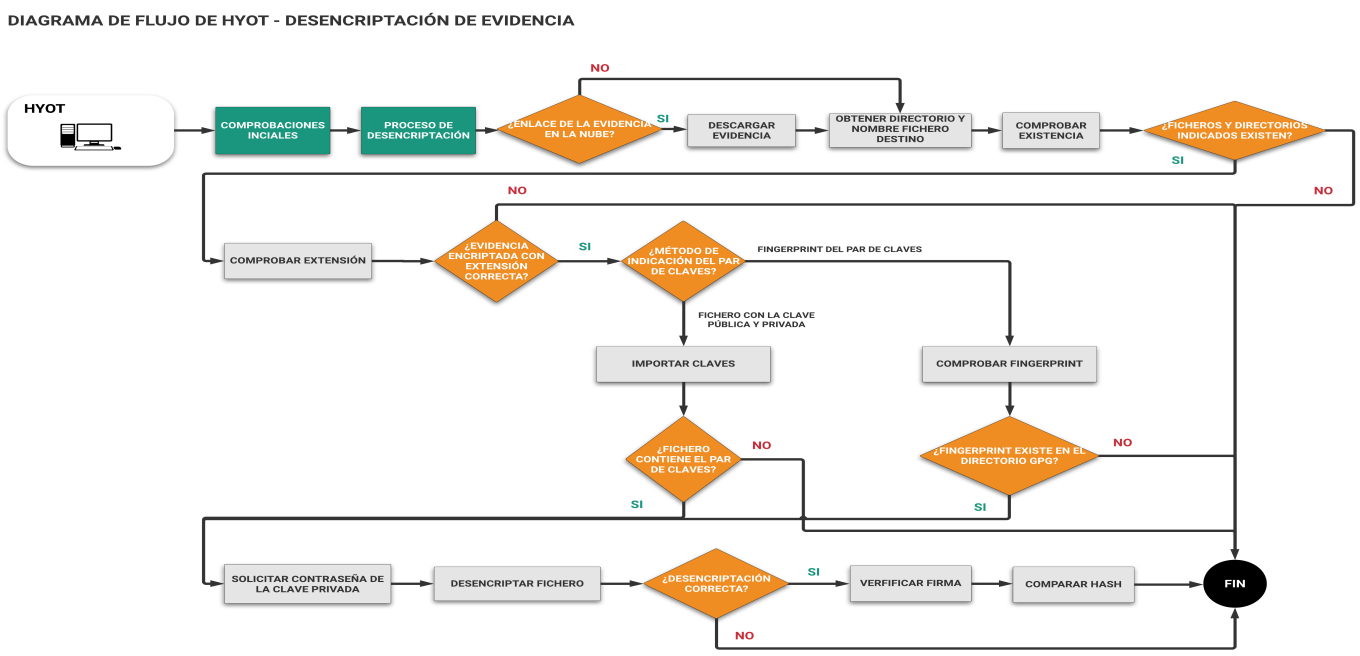


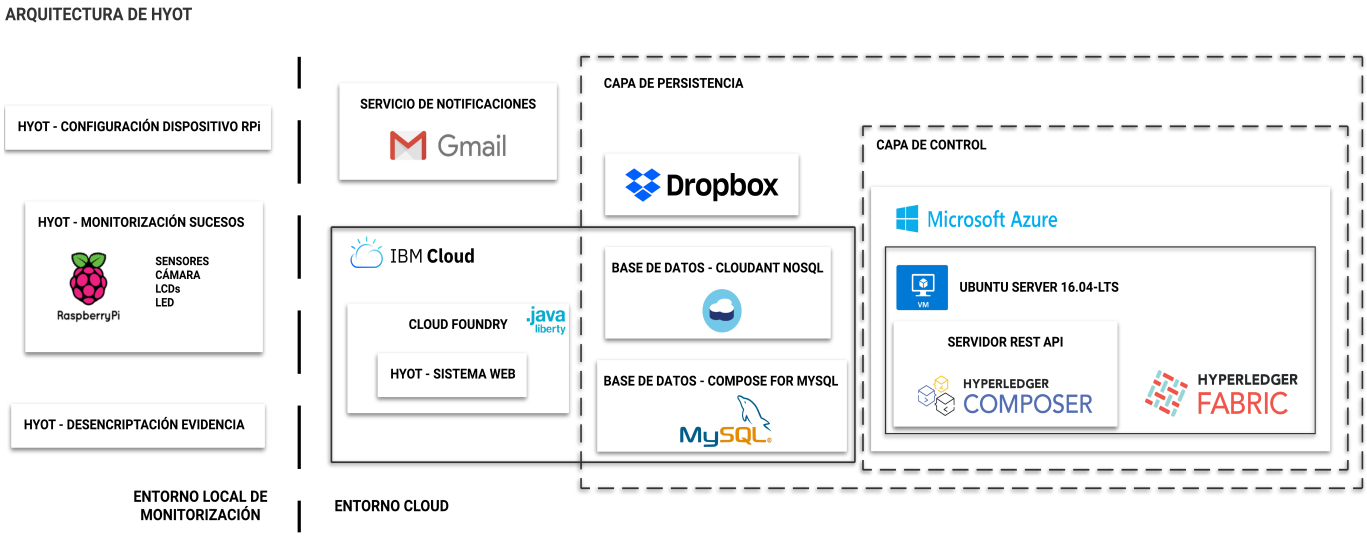
Figura 5.3: Diagrama de flujo - Desencriptación de evidencias.



5.2. Arquitectura de Hyot

La arquitectura de Hyot para la solución actual se compone de dos entornos y de diferentes servicios y plataformas. La Figura 5.4 detalla de forma más exhaustiva esta disposición enfatizando y concretando las herramientas empleadas y desplegadas.

Figura 5.4: Arquitectura de Hyot.



Observando la arquitectura se diferencian claramente dos entornos donde los diversos componentes están desplegados:

- Entorno de ámbito local donde se ubica el prototipo *hardware* elaborado, correspondiente a un cliente pesado. En él se despliega el componente de monitorización de sucesos, el componente de configuración del dispositivo RPi y el componente de desencriptación de evidencia, pudiendo este último ser desplegado en cualquier otro dispositivo siempre y cuando se disponga de todas las dependencias necesarias.

- Entorno en la nube donde se despliegan los dos componentes restantes: protocolo de registro de incidencias en la BC de HF y sistema web junto con los servicios empleados. Aquí, se puede diferenciar:

  - La capa de persistencia que engloba los cuatro mecanismos utilizados para persistir toda la información gestionada en el proyecto.
    - Por un lado, se encuentran las BBDD como servicio (*Database as a Service* -DBaaS-) Cloudant NoSQL DB y Compose for MySQL pertenecientes ambas al servicio IBM Cloud que mantienen el listado de mediciones y la gestión de usuarios y *token* del sistema web, respectivamente. Además, se encuentra el servicio de almacenamiento en la nube Dropbox (*Software as a Service* -SaaS-) [61].
    - Por otro lado y dentro de la capa de persistencia se encuentra la capa de control formada por la BC permisionada desplegada en HF la cual almacena alertas -información de incidencias-. Esta denominación se debe a que la información aquí guardada permite controlar y garantizar la veracidad de la trazabilidad registrada. Este despliegue se ha efectuado en una máquina virtual (*Virtual Machine* -VM-) Ubuntu en el servicio Microsoft Azure y en dicha máquina se localiza tanto HF como el servidor REST (*Representational State Transfer*) API (*Application Programming Interface*) de Hyperledger Composer (HC), siendo este servidor el que recibe peticiones securizadas por parte del componente de monitorización de sucesos y del sistema web y el que interactúa con la BC a través del nodo conectado.
  - El sistema web que actúa como cliente delgado para consumir la información monitorizada. Este sistema es desplegado en la plataforma como servicio (*Platform as a Service* -PaaS-) [61] *Liberty for Java* (Cloud Foundry) perteneciente al servicio IBM Cloud. Este tipo de despliegue garantiza, con respecto a uno local, una rápida provisión y configuración de la aplicación junto con una total disponibilidad.
  - El servicio para el envío de notificaciones mediante *emails*, siendo Gmail (SaaS) el configurado y utilizado.

5.3. Tecnologías de desarrollo

La codificación de los componentes de Hyot se ha llevado a cabo con diversas tecnologías de desarrollo. El empleo de varias, en concreto 4, se debe a las necesidades requeridas por cada componente ya sea por su grado de adaptación o porque se encuentre impuesto por la propia herramienta y a la experiencia de uso. A continuación, se indica cada una de ellas referenciando el componente o componentes donde se emplea y la justificación de ello:

- GNU Bash [25]: intérprete y procesador de comandos de *shell* que permite ejecutar *scripts* simples por consola con la finalidad de automatizar tareas que puede desempeñar un administrador de sistemas como puede ser la configuración de un sistema. Fue desarrollado para el proyecto GNU en el año 1987 como la versión libre del hasta entonces intérprete más importante de Unix: Bourne Shell (sh) y actualmente es el estándar de todas las distribuciones GNU/Linux. Esta tecnología es usada para codificar el componente de configuración del dispositivo RPi debido a que es la más idónea para automatizar las acciones de configuración inicial al no requerir demasiada complejidad en cuyo caso hubiese que haber empleado otros lenguajes de *scripting* más potentes como por ejemplo, Perl o Python.

- Javascript (JS) [47]: lenguaje de programación ligero e interpretado -orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico, entre otras características- que surge debido a la necesidad de ampliar las posibilidades de desarrollar sistemas web del lado del cliente (*front-end*) con contenido más complejo, dinámico y funcional ya que hasta entonces las prestaciones de estos sitios se encontraban limitadas exclusivamente a funcionalidades básicas sobre los elementos de texto, estilos y formatos, es decir, a contenido estático. Iniciado originalmente por Brendan Eich de la compañía de *software* Netscape Communications Corporation con el nombre de Mocha, el cual fue renombrado posteriormente a LiveScript, para finalmente quedar como JavaScript y continuado por la alianza entre Netscape y Sun, se ha popularizado enormemente hasta convertirse es un estándar basado en ECMAScript y soportado por todos los navegadores actuales. El uso de este lenguaje se ha centrado en el desarrollo de la red de negocio con la tecnología HC ya que es el lenguaje impuesto para describir la estructura de los recursos que participan en la BC. También, se ha empleado para definir la funcionalidad del lado de cliente del sistema web mediante la librería jQuery y otras librerías de funcionalidad específica.

- Python [69]: lenguaje de *scripting* de código abierto interpretado, interactivo y orientado a objetos creado por Guido van Rossum, preparado para desarrollar independientemente de la plataforma y sistema operativo desde aplicaciones de escritorio, *scripts* o incluso, sistemas web. En este tipo de lenguajes, que son más flexibles y portables que los lenguajes compilados, se emplea un programa intermedio llamado intérprete donde se traduce el código fuente a un pseudo código llamado *bytecode* la primera vez que se ejecuta, generando archivos `.pyc` o

`.pyo`, que son los que se ejecutarán en sucesivas ocasiones directamente en la computadora, en lugar de compilar el código a lenguaje máquina comprensible por este dispositivo. En los últimos años el lenguaje se ha hecho muy popular debido a todas las características que proporciona. Gracias a esto, a la facilidad de interacción con componentes electrónicos y al contexto y necesidades de desarrollo rápido y sencillo se ha empleado en:

  - El componente de monitorización de sucesos del entorno.
  - El componente de desencriptación de evidencia.

- Grails [26]: anteriormente llamado *Groovy on Rails*, surge en el año 2006 como respuesta a la necesidad de agilizar, automatizar y simplificar el desarrollo de aplicaciones web en JEE, evitando así el esfuerzo impuesto en la configuración inicial del entorno y proyecto. Se define como un *framework full-stack* -lado cliente y lado servidor- para aplicaciones web de propósito general, desarrollado sobre el lenguaje de programación Groovy (basado a su vez en el lenguaje Java), que permite la construcción de proyectos desde una filosofía ágil y sólida gracias al seguimiento de paradigmas como convención sobre configuración, *Don't repeat yourself* (DRY), *scaffolding* o inversión de control (IoC) y a la implementación del patrón de diseño Modelo-Vista-Controlador (MVC) donde prima minimizar el tiempo de configuración inicial y posterior desarrollo. Debido a todas estas características y a la experiencia de uso previa, esta tecnología se ha empleado para desarrollar el sistema web.

5.4. Red de negocio de Hyperledger Composer

La especificación de la red de negocio es un concepto clave en BC puesto que representa la digitalización de los elementos que la conforman junto con la lógica de negocio. En Hyot, se ha empleado la herramienta HC para la definición de esta red la cual es desplegada a través de un fichero `.bna` y mantenida por cada nodo conectado a la BC de HF[4]. Aunque HC no aprovecha aun todas las características que ofrece el *framework* HF, las necesidades actuales son cubiertas con creces con esta herramienta[5] y su uso se debe a los beneficios que proporciona:

- Simplificación del proceso de implementación de la BC ya que abstrae al desarrollador de numerosos procesos y conceptos los cuales son generados automáticamente a más bajo nivel al usar dicha herramienta.
- Aceleración de la creación de aplicaciones eliminando el esfuerzo requerido si se hiciese a un más bajo nivel de codificación -utilizando directamente HF-.

[4]En el contexto actual del Trabajo Fin de Máster (TFM) y debido a las limitaciones existentes, la red está formada por un único nodo, en concreto, aquel desplegado en HF de la VM establecida en el servicio Microsoft Azure.

[5]HC se encuentra bajo continuo desarrollo y está respaldado por una fuerte comunidad de desarrolladores, hecho contrastado por la liberación constante de nuevas *releases*. Consultar [34] para obtener más información.

- Posibilidad de prueba de la definición de la red realizada antes de ser desplegada en un entorno de producción lo que ayuda a reducir riesgos.

La solución de la red de negocio construida es muy simple y se detalla a continuación para un pleno entendimiento. En primer lugar, se van a definir los conceptos claves que conforman una red los cuales corresponden a definiciones de clases instanciables:

- Activo o *asset*: hace referencia a los bienes, servicios o propiedades tanto tangibles como intangibles que son almacenados en registros en la BC. Dicho de otra forma, es cualquier cosa del mundo real que pueda ser representado y utilizado en una red de negocio.
- Participante o *participant*: representa los miembros de una red de negocio los cuales poseen activos y envían transacciones para intercambiarlos o para modificar sus propiedades.
- Transacción o *transaction*: simbolizan las reglas de negocio permitidas, es decir, los procedimientos por el cual los participantes interactúan con los activos.

En el dominio de nuestro caso de uso -versión actual de la red 1.1.2- se ha determinado el siguiente espacio de nombres `org.hyot.network.` y se ha definido únicamente un tipo de activo (estereotipo `«asset»`) con nombre `Alert` que representa una incidencia en el entorno monitorizado, es decir, la alerta de un suceso anómalo y contiene un dato clave de la evidencia, valor *hash*. Este activo, que ha sido especificado utilizando a mayores dos tipos enumerados y una clase abstracta[6] (estereotipo `«concept»`[7]) como objetivo meramente de aprendizaje[8], se compone de:

- Identificador de la alerta (tipo `String`) que actúa como clave primaria, es decir, valor único. Coincide con el identificador de la medición almacenada en la BBDD a la que está asociada.
- Clase `AlertDetails` con estereotipo `«concept»` que contiene los detalles de la alerta[9]:
  - Marca temporal (*timestamp*) en la que ocurre el suceso (tipo `DateTime`).
  - Sensor que origina el protocolo de alerta de tipo `SensorType` el cual representa una enumeración.
  - Evento del sensor que origina el protocolo de alerta de tipo `EventType` el cual representa una enumeración.

[6]Indica que el tipo no puede ser instanciado.

[7]Clase que no representa ni un activo, ni un participante ni una transacción y típicamente son contenidos por alguno de ellos.

[8]El foco también ha sido aprender la tecnología por lo que se han definido entidades y propiedades adicionales o de la forma menos directa.

[9]El valor de cada propiedad aquí almacenada, con fin estadístico, coincide con el almacenado en la medición -excepto el valor *hash* que no es registrado en la medición-. Sí que es verdad, que al presentar el mismo identificador se podría relacionar la medición y alerta y obtener la información pero requiere de una complejidad mayor en la lógica.

- Valor *hash* del contenido de la evidencia original -sin encriptar ni firmar- (tipo `String`).
- Enlace al servicio de almacenamiento en la nube donde se ubica la evidencia encriptada y firmada (tipo `String`).
- Participante de tipo `User` que ejecuta la transacción de publicación de una nueva alerta (activo `Alert`).

Cada enumerado contiene a su vez los posibles valores para las propiedades que hacen referencia al sensor y evento que lanzan la alerta, siendo éstos los siguientes:

- Enumerado `SensorType`: DHT11 y HCSR04.
- Enumerado `EventType`: TEMPERATURE, HUMIDITY, DISTANCE.

Con respecto a los participantes se ha definido al participante `User` -estereotipo `«participant»`- que representa el usuario o dispositivo que registra una alerta en la BC y contiene las siguientes propiedades:

- Nombre de usuario que actúa como clave primaria debiendo ser único su valor (tipo `String`).
- Dirección de correo electrónico el cual es una propiedad opcional (tipo `String`).
- Nombre real (tipo `String`).
- Apellidos (tipo `String`).

La lógica de negocio está definida por la especificación de la transacción `PublishAlert` (estereotipo `«transaction»`) que contiene los mismos campos que los del activo `Alert`. Esta transacción es ejecutada por una función procesadora de transacciones cuyo propósito es el registro de un nuevo activo `Alert` en la BC. Además, en la red de negocio se pueden especificar consultas (*querys*) que se ejecutan contra la BC y definir los privilegios de acceso para los participantes. Para el primer caso, se han definido las siguientes consultas algunas de ellas parametrizadas[10].

- Seleccionar todas las alertas.
- Seleccionar las alertas generadas por un determinado sensor.
- Seleccionar las alertas generadas por un determinado evento.
- Seleccionar las alertas generadas por un determinado sensor y registradas por un usuario específico.

[10]Cabe indicar que la obtención de la información también puede realizarse mediante llamadas directas a la entidad, aunque de esta forma únicamente se puede obtener información simple y no filtrada. Además, con el formato de consulta se evita la dependencia de indicar el *namespace* de la red.

- Seleccionar las alertas generadas por un determinado evento y registradas por un usuario específico.
- Seleccionar todos los usuarios.
- Seleccionar el usuario con un identificador concreto.
- Seleccionar las alertas registradas por un usuario específico.

Con respecto a la lista de control de acceso, ésta contiene las siguientes reglas:

- Todos los participantes de tipo `User` pueden registrar transacciones `PublishAlert`.
- Todos los participantes de tipo `User` pueden crear activos de tipo `Alert` cuando se registren transacciones `PublishAlert`. Esto quiere decir que este tipo de activos no pueden ser creados directamente con este tipo de participante.
- Todos los participantes de tipo `User` pueden obtener el registro de los activos `Alert` cuyo poseedor sean ellos mismos lo que implica que los registros una vez almacenados no puedan ser modificados ni eliminados, proporcionando así una capa extra de seguridad al asegurar que si una alerta fue almacenada es porque realmente sucedió. Esta regla implica también que un determinado participante `User` no puede consultar los activos `Alert` registrados por otros participantes de este tipo.
- Los administradores de la red de negocio presentan acceso completo a los recursos de usuario por lo que pueden realizar directamente cualquier tipo de operación sobre las entidades participante, activo y transacción lo cual es denegado al participante de tipo `User` en algunos casos.
- Los administradores de la red de negocio presentan acceso completo a los recursos de sistema lo que permite por ejemplo, desplegar la red de negocio, gestionar identidades, etc.

Hay que tener en cuenta que las reglas definidas son evaluadas en orden de definición (arriba-abajo) de tal manera que cuando una acción cumple una determinada regla, ésta se aplica y el resto de reglas son ignoradas.

5.5. Jerarquía de ficheros

En esta sección se desglosa la estructura de directorios y ficheros del proyecto -según la jerarquía completa situada bajo el directorio llamado `Hyot-` diferenciando los componentes que lo conforman:

- `.gitignore`: fichero oculto que permite excluir del control de versiones Git determinados archivos los cuales son ignorados y por tanto no versionados en el repositorio de código.

- `README.md`: fichero de texto en formato markdown que documenta el proyecto Hyot, es decir, contiene información acerca de éste como la descripción, componentes, dependencias, guía de instalación y de uso, autor, etc.

- **hyot_app:** componente del proyecto Hyot referido al sistema web desarrollado con el *framework* Grails para consumir la información monitorizada[11].

  - `build`: directorio que contiene el código compilado junto con ficheros propios de la aplicación web.
  - `grails-app`: directorio que contiene todos los artefactos Groovy que el entorno genera y que el desarrollador debe modificar para implementar la funcionalidad.
    - `assets`: directorio que contiene las hojas de estilo CSS (*Cascading Style Sheets*) y los ficheros JS. Estos ficheros también pueden ser situados bajo el directorio `src/main/webapp`. Sin embargo, en el directorio actual son tratados de manera más eficiente.
    - `conf`: contiene ficheros de configuración de la aplicación.
      - `spring`: directorio que contiene el fichero `resources.groovy` donde se registran e inyectan *beans* -objetos adicionales- usados por Spring DSL.
      - `application.groovy`: fichero que contiene la configuración del *plugin* Spring Security Core.
      - `application.yml`: fichero YAML (*YAML Ain't Markup Language*) que contiene las principales configuraciones como, por ejemplo la BBDD, las credenciales del servicio de envío de notificaciones, etc.
      - `logback.groovy`: fichero que contiene la configuración para el *logger*.
    - `controllers`: directorio que contiene los controladores de la aplicación.
      - `ControlPanel`: directorio que contiene los controladores empleados por el usuario administrador.
      - `CustomUserTasks`: directorio que contiene el controlador que implementa la funcionalidad para un usuario sin registrar (redirección tras autenticación, autenticación fallida, restablecer contraseña, etc.).
      - `grails/plugin/springsecurity`: directorio que contiene el controlador definido por el *plugin* Spring Security Core para el procedimiento de autenticación. Este controlador por defecto se ubica en el directorio del propio *plugin*. Sin embargo, se ha ubicado aquí porque se han redifinido determinadas acciones.
      - `hyot_app`: directorio que contiene el fichero `UrlMappings.groovy` el cual define el mapeo de las direcciones URLs (*Uniform Resource Locator*).

[11] Únicamente se especifican los directorios y ficheros más importantes.

        - ◇ `Security`: directorio que contiene el controlador para gestionar la entidad `SecUser` la cual está referida al usuario administrador.
        - ◇ `User`: directorio que contiene el controlador para gestionar la entidad `User` la cual está referida al usuario normal.
        - ◇ `UserPage`: directorio que contiene el controlador empleado por el usuario normal.
    - `domain`: directorio que contiene las clases de dominio mapeadas como tablas en la BBDD, donde se ubican las entidades referentes al usuario administrador, al usuario normal, a los roles, relaciones entre éstos y al *token*.
    - `i18n`: directorio que contiene los ficheros de mensajes de internacionalización de la aplicación que permiten adaptar el idioma de acuerdo al lenguaje del navegador.
    - `init`: directorio que contiene el fichero `BootStrap.groovy` que especifica las acciones a ejecutar al arranque y/o destrucción de la aplicación.
    - `services`: directorio que contiene las clases que modelan los servicios. Corresponde con la capa de servicio.
        - ◇ `Certificate`: directorio que contiene el servicio para saltar la verificación del certificado al hacer peticiones al servidor de HC.
        - ◇ `Cloudant`: directorio que contiene el servicio que maneja la lógica de Cloudant NoSQL DB.
        - ◇ `CustomUserTasks`: directorio que contiene el servicio que maneja la lógica de las funcionalidades de un usuario sin registrar.
        - ◇ `HyperledgerFabric`: directorio que contiene el servicio que maneja la lógica de las peticiones a la BC de HF a través de HC.
        - ◇ `Security`: directorio que contiene el servicio para gestionar los *tokens*.
    - `taglib`: directorio que contiene las librerías de etiquetas que el desarrollador crea para ser utilizadas en las vistas.
    - `views`: directorio que contiene ficheros GSP (*Groovy Server Pages*) que modelan las vistas de la aplicación.
- `src`: directorio que contiene la especificación de tests y otras clases Java y/o Groovy. Entre esta clases, se han definido algunas para extender la funcionalidad o gestionar las sesiones concurrentes, la redirección tras una autenticación exitosa, el manejo de cierre de sesión, etc.JS
- `build.gradle`: fichero que contiene distintos aspectos del proceso de compilación como las rutas de trabajo y la resolución de dependencias.

- **hyot_bnd:** componente del proyecto Hyot que contiene la especificación de la red de negocio desplegada en la BC de HF. El desarrollo de este componente se ha realizado con HC.

- `dist`: directorio donde se almacena la red de negocio generada (`hyot-network@version.bna`).
- `doc`: directorio donde se almacena la documentación generada con el lenguaje de marcado JSDoc (*JavaScript Doc*).
- `lib`: directorio donde se ubican los ficheros que permiten implementar los requisitos de funcionalidad de la red de negocio.
  - `hyot_logic.js`: fichero JS que define la lógica de las funciones procesadoras de transacciones para la red de negocio.
- `models`: directorio donde se ubican los ficheros que permiten modelar la estructura y dominio de una red de negocio (activos, participantes y transacciones) y las relaciones entre los elementos.
  - `hyot_model.cto`: fichero que modela la red de negocio.
- `node_modules`: directorio que contiene los paquetes de Node.js instalados localmente al proyecto con la herramienta NPM (*Node Package Manager*).
- `test`: directorio que contiene la especificación de los tests unitarios.
  - `hyot_unit.js`: fichero que contiene test unitarios sobre la red de negocio especificada.
- `uml`: directorio que contiene diagramas UML (*Unified Modeling Language*) de los modelos de la red de negocio, generados con la herramienta PlantUML[12].
  - `hyot_bnd_uml.png`: imagen del diagrama UML del modelo de Hyot.
  - `hyot_bnd_uml.puml`: fichero PlantUML del modelo de Hyot.
- `.eslintignore`: fichero asociado a la herramienta *open source* ESLint que permite excluir del proceso del *linter* determinados ficheros los cuales son ignorados.
- `.eslintrc.yml`: fichero en formato YAML que especifica la configuración de la herramienta ESLint.
- `jsdoc.json` : fichero en formato JSON (*JavaScript Object Notation*) que contiene la configuración para la generación de documentación con JSDoc.
- `package.json`: fichero JSON donde se especifican metadatos de identificación del proyecto (nombre, versión, descripción, repositorio, autores, licencia, etc.) y las dependencias necesarias para su funcionamiento.
- `package-lock.json`: fichero JSON que permite mantener el detalle específico de las versiones de dependencias que están instaladas en el proyecto de forma que garantiza que todos los colaboradores presenten las mismas versiones de paquetes.

[12]Extensión para la herramienta Visual Studio Code que requiere tener instalados las herramientas Java y Graphviz. Para obtener más información, consultar [53].

- `permissions.acl`: fichero que detalla las reglas de control de acceso para los participantes de la red de negocio.
- `queries.qry`: fichero que contiene la definición de las consultas que se pueden realizar contra la BC desplegada.
- `README.md`: fichero de texto en formato markdown que documenta la red de negocio.

- **hyot_raspberry/hyot:** componente del proyecto Hyot, formado por ficheros Python, que monitoriza sucesos -eventos temperatura, humedad y distancia- del entorno IoT.
  - `logs`: directorio donde se almacena el fichero de *log* generado durante la ejecución de este componente. Este fichero contiene la salida por consola.
  - `conf`: directorio de configuración.
    - `hyot.yml`: fichero YAML que define la configuración de determinados servicios (Gmail, Dropbox, etc.). Este fichero no está bajo control de versiones y es de obligatoriedad su existencia.
  - `template`: directorio que contiene las plantillas para las notificaciones mediante *emails*.
    - `email_template.html`: plantilla HTML (*HyperText Markup Language*) referida al envío de una notificación de *email* cuando se produce una alerta.
    - `error_measurement_template.html`: plantilla HTML referida al envío de una notificación de *email* cuando se produce algún error durante la medición.
  - `camera_module.py`: módulo que contiene la lógica para manejar el dispositivo Picamera.
  - `checks_module.py`: módulo que parsea el menú comprobando las opciones introducidas por el usuario y realiza diversas comprobaciones iniciales tales como la comprobación de ejecución con usuario superprivilegiado, comprobación de que se está ejecutando en una plataforma GNU/Linux y en un dispositivo RPi, etc.
  - `cloudantdb_module.py`: módulo que gestiona la lógica con la plataforma Cloudant NoSQL DB del servicio IBM Cloud.
  - `dropbox_module.py`: módulo que gestiona la funcionalidad con el servicio Dropbox.
  - `email_module.py`: módulo que gestiona el envío de *emails* a través del servicio Gmail.
  - `gpg_module.py`: módulo que hace uso de la funcionalidad provista por GPG para la encriptación y firmado de evidencias.
  - `hyot_main.py`: fichero que contiene la funcionalidad principal para efectuar la monitorización del entorno. Este fichero importa todos los módulos y realiza llamadas a las funciones de éstos.
  - `hyperledgerFabric_module.py`: módulo que gestiona la interacción con la BC de HF.

- `lcd_module.py`: módulo que maneja los LCDs.
- `logger.py`: clase que permite redirigir la salida de la ejecución tanto a un fichero de *log* como a la consola.
- `system_module.py`: módulo que ejecuta funciones sobre el sistema local, en concreto, la creación, eliminación y establecimiento de permisos de directorios y/o ficheros y la comprobación de existencia de ficheros.
- `token_module.py`: módulo que contiene funciones para generar *tokens* empleados en la securización del servidor REST API de HC.

- **hyot_raspberry/hyot_decryption:** componente del proyecto Hyot, formado por dos ficheros Python, que permite la desencriptación de la evidencia previamente encriptada y firmada con GPG.
  - `hyot_decryption.py`: fichero que contiene la funcionalidad principal para efectuar la desencriptación, la verificación de la firma y la comparación del valor *hash* entre otras comprobaciones.
  - `menu_module.py`: fichero que parsea el menú y comprueba las opciones introducidas por el usuario.
- **hyot_raspberry/setup:** componente del proyecto Hyot, formado por dos ficheros Bash (*Bourne-again Shell*), que permite configurar el dispositivo RPi.
  - `raspberrypi_setup.sh`: fichero que ejecuta una serie de comprobaciones iniciales, instala las dependencias y habilita las interfaces necesarias.
  - `utils.sh`: fichero que contiene funciones para proporcionar formato y estilo al texto mostrado por consola.

5.6. Mapeo de URLs

Las direcciones URL son uno de los mecanismos por el cual el usuario puede interactuar con el sistema o la API expuesta por éste. Aunque en el proyecto actual el usuario no debe emplearlas de una forma directa y son totalmente transparentes para éste, en esta sección se exponen como punto meramente informativo.

5.6.1. Servidor REST API de Hyperledger Composer

El servidor expuesto por la tecnología HC genera una REST API establecida dinámicamente al momento de desplegar la red de negocio en base a su especificación, abstrayendo del bajo nivel de interacción al usuario y permitiendo interactuar con la BC de una manera directa y sencilla

mediante los protocolos HTTP (*Hypertext Transfer Protocol*) o HTTPS (*Hypertext Transfer Protocol Secure*)[13].

Para el proyecto Hyot se ha adoptado una política de seguridad estricta por lo que únicamente se pueden realizar a este servidor peticiones a través de una comunicación segura. De esta forma, es requisito indispensable para el funcionamiento correcto que el servidor sea arrancado de forma segura lo que conlleva a su vez la generación de un certificado. Este servidor expone la red de negocio en la dirección: `https://localhost:3000/explorer`[14]. A esta dirección se la debe añadir el espacio de nombres definido `-org.hyot.network.-` en la red de negocio a la hora de realizar cualquier solicitud a uno de los tres elementos que forman el modelo (*asset*, *transaction* o *participant*). Además, el servidor expone el acceso a otros elementos como son las consultas (queries), el historian o las identidades (identities). A continuación, se muestra el mapeo de direcciones de este servidor.

```
| MAPEO DE URLS DEL SERVIDOR REST API DE HYPERLEDGER COMPOSER
| URL base: /api
| Versión de la API: 1.0.0
| ------------------------------------------------------

# Asset: Alert
| GET    | /org.hyot.network.Alert         | Acción: Retorna todas las instancias Alert               |
| POST   | /org.hyot.network.Alert*        | Acción: Crea una nueva instancia Alert                   |
| GET    | /org.hyot.network.Alert/{id}    | Acción: Retorna la instancia Alert que posea ese id      |
| HEAD   | /org.hyot.network.Alert/{id}    | Acción: Comprueba si una instancia Alert existe          |
| PUT    | /org.hyot.network.Alert/{id}*   | Acción: Remplaza propiedades de una instancia Alert      |
| DELETE | /org.hyot.network.Alert/{id}    | Acción: Borra la instancia Alert que posea ese id        |

 # Transaction: PublishAlert
| GET    | /org.hyot.network.PublishAlert  | Acción: Retorna todas las instancias de PublishAlert     |
| POST   | /org.hyot.network.PublishAlert* | Acción: Crea una nueva instancia de PublishAlert         |
| GET    | /org.hyot.network.PublishAlert/{id} | Acción: Retorna la instancia de PublishAlert         |

# Participant: User
| GET    | /org.hyot.network.User          | Acción: Retorna todas las instancias User                |
| POST   | /org.hyot.network.User*         | Acción: Crea una nueva instancia User                    |
| GET    | /org.hyot.network.User/{id}     | Acción: Retorna la instancia User que posea ese id       |
| HEAD   | /org.hyot.network.User/{id}     | Acción: Comprueba si una instancia User existe           |
| PUT    | /org.hyot.network.User/{id}*    | Acción: Remplaza propiedades de una instancia User       |
| DELETE | /org.hyot.network.User/{id}     | Acción: Borra la instancia User que posea ese id         |

# Queries
| GET    | /queries/Alert                          | Acción: Retorna todas las instancias Alert       |
| GET    | /queries/AlertFromSpecificEvent         | Acción: Retorna todas las instancias Alert
    lanzadas por un determinado evento                                                          |
| GET    | /queries/AlertFromSpecificEventAndUser  | Acción: Retorna todas las instancias Alert
    lanzadas por un determinado sensor y registradas por un User concreto                       |
```

[13]Otra método de interacción es utilizar directamente HF. Sin embargo, el mismo proceso es más complejo al tratarse de un más bajo nivel..

[14]Esta dirección para el despliegue actual realizado es: `https://40.121.15.202:3000/explorer`.

```
| GET    | /queries/AlertFromSpecificSensor        | Acción: Retorna todas las instancias Alert
    lanzadas por un determinado sensor                                                    |
| GET    | /queries/AlertFromSpecificSensorAndUser | Acción: Retorna todas las instancias Alert
    lanzadas por un determinado sensor y registradas por un User concreto                 |
| GET    | /queries/AlertsOfUser                   | Acción: Retorna todas las instancias Alert
    registradas por un User concreto                                                      |
| GET    | /queries/User                           | Acción: Retorna todas las instancias User       |
| GET    | /queries/User                           | Acción: Retorna la instancia User con un
    determinado identificador                                                             |

# System: Métodos generales de la red de negocio
| GET    | /system/historian               | Acción: Retorna todos los registros del historian       |
| GET    | /system/historian{id}           | Acción: Retorna el registro del historian con ese id    |
| GET    | /system/identity}               | Acción: Retorna todas las identidades                   |
| GET    | /system/identity{id}            | Acción: Retorna la identidad que posea ese id           |
| POST   | /system/identity{id}/revoke     | Acción: Revoca la identidad especificada                |
| POST   | /system/identity/bind*          | Acción: Enlaza una identidad a un participante existente |
| POST   | /system/identity/issue*         | Acción: Emite una identidad al participante especificado |
| GET    | /system/ping                    | Acción: Prueba la conexión con la red de negocio        |
```

* Estas consultas a la API requieren de información adicional introducida como parámetro en formato *JSON*. Existen otras peticiones que deben incluir un único parámetro como, por ejemplo: el identificador o el nombre del sensor o evento. Sin embargo, no requieren de una estructura de datos en este formato por lo que se omiten a continuación.

La estructura de la información de cada petición para la red de negocio de Hyot se puede consultar directamente en la dirección web donde se expone la API -citada anteriormente-. Aun así, para evitar que el lector tenga que desplegar este servicio, a continuación se presenta la estructura de datos necesaria para cada petición que deba incluir algún parámetro.

```
| ESTRUCTURA DE PARÁMETROS EN FORMATO JSON
| Versión de la red de negocio Hyot: 1.0.0
| ----------------------------------------

# Asset: Alert - Method: POST, PUT
{
  ``$class": ``org.hyot.network.PublishAlert",
  ``alert_id": ``string",
  ``alert_details": {
     ``$class": ``org.hyot.network.AlertDetails",
     ``timestamp": ``datetime (e.g. 2018-07-01T06:08:20.001Z)",
     ``sensor_origin": ``DHT11|HCSR04",
     ``event_origin": ``TEMPERATURE|HUMIDITY|DISTANCE",
     ``hash": ``string",
     ``link": ``string",
     ``owner": ``resource:org.hyot.network.User#identifier"
  }
```

```
}

# Transaction: PublishAlert - Method: POST
{
  ``$class": ``org.hyot.network.PublishAlert",
  ``alert_id": ``string",
  ``alert_details": {
     ``$class": ``org.hyot.network.AlertDetails",
     ``timestamp": ``datetime (e.g. 2018-07-01T06:08:20.001Z)",
     ``sensor_origin": ``DHT11|HCSR04",
     ``event_origin": ``TEMPERATURE|HUMIDITY|DISTANCE",
     ``hash": ``string",
     ``link": ``string",
     ``owner": ``resource:org.hyot.network.User#identifier"
  }
}

# Participant: User - Method: POST, PUT
{
  ``$class": ``org.hyot.network.User",
  ``username": ``string",
  ``email": ``string",
  ``first_name": ``string",
  ``last_name": ``string"
}

# System: Identities Bind - Method: POST
{
  ``participant": ``string",
  ``certificate": ``string"
}

# System: Identities Issue - Method: POST
{
  ``participant": ``string",
  ``userID": ``string",
  ``options": {}
}
```

5.6.2. Sistema web

En esta sección se presentan las diferentes direcciones URLs mapeadas en acciones de controladores. También se debe indicar que existen más acciones en algunos controladores que no han sido mapeadas explícitamente en una dirección debido a que se trata de llamadas internas, como por ejemplo: llamadas AJAX o acción de eliminación de una instancia de una entidad (método

HTTP delete, etc.).

```
| MAPEO DE URLS DEL SISTEMA WEB
| Dirección del despliegue actual: https://hyot.eu-gb.mybluemix.net/
| ------------------------------------------------------------

# Métodos de usuario sin registrar
###################################
# Controlador: login
|  *  | /                                                | Acción: auth                                |
# Controlador: customUserTasks
|  *  | /login/loggedIn                                  | Acción: loggedIn                            |
|  *  | /authFail                                        | Acción: authFail                            |
|  *  | /forgotPassword                                  | Acción: restorePassword                     |
|  *  | /newPassword                                     | Acción: changePass                          |

# Métodos de usuario administrador
###################################
# Controlador: controlPanel
|  *  | /dashboard                                       | Acción: dashboard                           |
# Controlador: alert
|  *  | /alert                                           | Acción: getAllAlerts                        |
# Controlador: measurement
|  *  | /measurement                                     | Acción: getAllMeasurements                  |
# Controlador: secUser
|  *  | /administrator                                   | Acción: index                               |
|  *  | /administrator/create                            | Acción: create                              |
|  *  | /administrator/create-error                      | Acción: save                                |
|  *  | /administrator/edit/${id}?(.${format)?           | Acción: edit                                |
|  *  | /administrator/edit-error/${id}?(.${format)?     | Acción: update                              |
|  *  | /administrator/edit/profileImage/${id}?(.${format)?        | Acción: editProfileImage          |
|  *  | /administrator/edit-error/profileImage/${id}?(.${format)?  | Acción: updateProfileImage        |
# Controlador: user
|  *  | /user                                            | Acción: index                               |
|  *  | /user/create                                     | Acción: create                              |
|  *  | /user/create-error                               | Acción: save                                |
|  *  | /user/edit                                       | Acción: edit                                |
|  *  | /user/edit-error/${id}?(.${format)?              | Acción: update                              |
|  *  | /user/edit/profileImage/${id}?(.${format)?       | Acción: editProfileImage                    |
|  *  | /user/edit-error/profileImage/${id}?(.${format)?  | Acción: updateProfileImage                 |

# Métodos de usuario normal
############################
# Controlador: userPage
|  *  | /home                                            | Acción: home                                |
|  *  | /profile                                         | Acción: profile                             |
|  *  | /profile-error                                   | Acción: updatePersonalInfo                  |
|  *  | /profilePassword                                 | Acción: profilePassword                     |
|  *  | /profilePassword-error                           | Acción: profilePassword                     |
|  *  | /profileAvatar                                   | Acción: profileAvatar                       |
|  *  | /profileAvatar-error                             | Acción: updateAvatar                        |
|  *  | /mymeasurements                                  | Acción: myMeasurements                      |
|  *  | /myalerts                                        | Acción: myalerts                            |

# Métodos generales
|  *  | /${controller}/${action}?/${id}?(.${format)?     | Acción: (acción por defecto)                |
|  *  | /noRole                                          | Vista:   noRole                             |
```

```
|  *  | ERROR: 400                                  | Vista:   /error/badRequest                |
|  *  | ERROR: 401                                  | Vista:   /error/unauthorized              |
|  *  | ERROR: 403                                  | Vista:   /error/denied                    |
|  *  | ERROR: 404                                  | Vista:   /error/notFound                  |
|  *  | ERROR: 405                                  | Vista:   /error/notAllowed                |
|  *  | ERROR: 500                                  | Vista:   /error/internalError             |
|  *  | ERROR: 503                                  | Vista:   /error/unavailableService        |
|  *  | /humans.txt                                 | Vista:   extraInformation/humans          |
|  *  | /robots.txt                                 | Vista:   extraInformation/robots          |
```

5.7. Seguridad

Otro aspecto donde se ha intentado poner énfasis es la seguridad ya que hoy en día en muchos desarrollos es obviada parcialmente. Una buena seguridad evita todo tipo de contratiempos presentes y futuros. Para ofrecer seguridad en Hyot se han securizado las comunicaciones con los servidores empleados: HC y sistema web. Además, para este último se ha empleado el *plugin* Spring Security Core el cual gestiona todos los procesos relativos a la autenticación de los usuarios y a la autorización de éstos mediante permisos, como por ejemplo: protección de direcciones, autenticación básica y avanzada, definición de roles y control de acceso, securización de métodos y usuarios de la capa de negocio, restricción de acceso basada en IP, etc. La red de negocio desplegada en la BC también dispone por su parte de una lista de control de acceso para limitar quién puede acceder y a qué recursos.

5.7.1. Securización de servidores

En el proyecto existen dos servidores que reciben peticiones desde un agente externo y retornan respuestas en base a esas solicitudes. En estos servidores (servidor de HC y servidor del sistema web) la comunicación podría ser realizada a través del protocolo HTTP. Esta comunicación que es completamente funcional, presenta el problema de garantizar la seguridad puesto que cualquier dato se transmite en texto plano -sin cifrar- y por tanto cualquier intruso que tenga acceso a la red puede capturar y analizar los datos intercambiados, desembocando en un fallo de seguridad grave cuando se trata de datos sensibles como, por ejemplo contraseñas o *tokens* de autenticación.

Por esta razón, hoy en día la gran mayoría de sistemas que tratan datos sensibles implementan el protocolo HTTPS con el fin de ofrecer conexiones seguras e impedir que intrusos puedan conocer de una manera directa estos datos y los movimientos realizados. Sobre esta conexión, los datos navegan de manera cifrada y solamente los poseedores de las claves de cifrado pueden leer su contenido. Aun así, esta medida no es suficiente ya que puede existir otro problema adicional en la comunicación: la falsedad de la identidad del servidor. Dicho esto, otra consideración es la implementación de la autenticación de servidores mediante certificados SSL para confirmar que el servidor con el que se está estableciendo la comunicación es quién dice ser. Esta autenticación

es realmente vital ya que el proceso de generación de certificados es público y cualquiera con las condiciones necesarias (posesión de claves) puede falsear la información.

Para solucionar este problema existen además las autoridades de confianza[15] (*Certificate Authority* -CA-) las cuales se dedican a asegurar que un certificado es válido y el dominio indicado pertenece al propietario. Estos certificados aseguran que los datos son enviados al servidor correcto pero previamente entre el cliente y el servidor debe ocurrir una fase de negociación (protocolo *handshake* o apretón de manos) de los detalles técnicos (versión del protocolo, algoritmos de cifrado que se usarán, etc.) usados en la comunicación. Esta fase se puede resumir en los siguientes puntos:

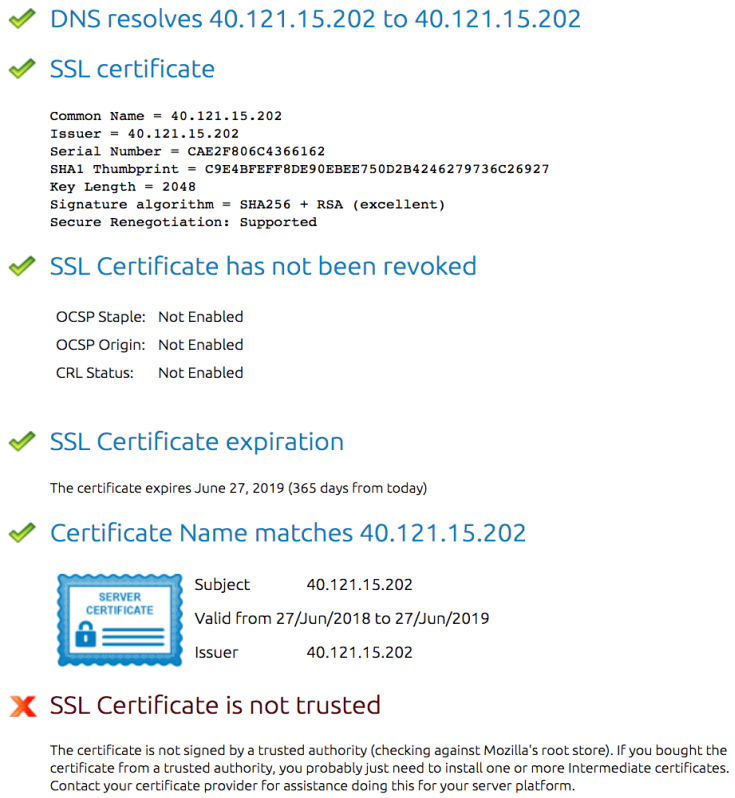
1. El cliente intenta conectarse al servidor o sitio web protegido con una comunicación segura.
2. El cliente solicita que el servidor se identifique.
3. El servidor envía al cliente una copia de su certificado.
4. El cliente comprueba si confía en el certificado. De ser así, envía un mensaje al servidor.
5. El servidor reenvía un reconocimiento firmado digitalmente para iniciar una sesión cifrada con SSL (*Secure Sockets Layer*).
6. Los datos cifrados se comparten entre el cliente y el servidor.

En Hyot se ha impuesto el uso de comunicaciones seguras. Debido al carácter del trabajo y al coste que presentan los certificados firmados por una CA se ha utilizado un certificado autofirmado. La única diferencia que presentan este tipo de certificados es que son firmados por uno mismo con la clave privada generada previamente. Debido a ello, su uso únicamente es recomendado para situaciones de desarrollo y/o pruebas. Este certificado que debe ser desplegado en ambos servidores[16] habilita la comunicación segura y con el fin analizar de forma superficial la seguridad, a continuación se presenta una serie de pruebas que se han realizado en este caso sobre el servidor de HC. La Figura 5.5 muestra el análisis realizado con una herramienta *online* donde se observa información sobre el certificado, la fecha de validez, el algoritmo de firma utilizado, si el certificado es confiable o no, etc. En este caso, el certificado es marcado como no confiable porque es de tipo autofirmado, sin embargo se trata de un aviso ya que el funcionamiento es correcto por lo que no afecta a la seguridad. La Figura 5.6 muestra la advertencia si se hubiese empleado el algoritmo de firma por defecto SHA-1 el cual se considera inseguro hoy en día.

[15]Puede consultar el listado de autoridades en [9].

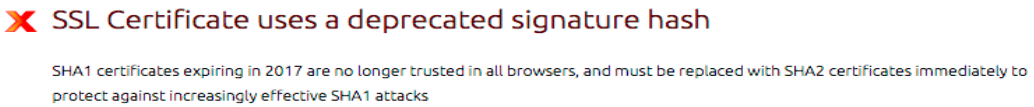
[16]En el caso del sistema web, al ser desplegado en un servicio de IBM Cloud no ha hecho falta su instalación para habilitar la comunicación segura ya que por defecto se utilizan los certificados de dicha plataforma.

Figura 5.5: Análisis de seguridad del certificado.



Fuente de comprobación: `https://www.digicert.com/help/`

Figura 5.6: Análisis de seguridad del certificado - Algoritmo de firma SHA-1.



Fuente de comprobación: `https://www.digicert.com/help/`

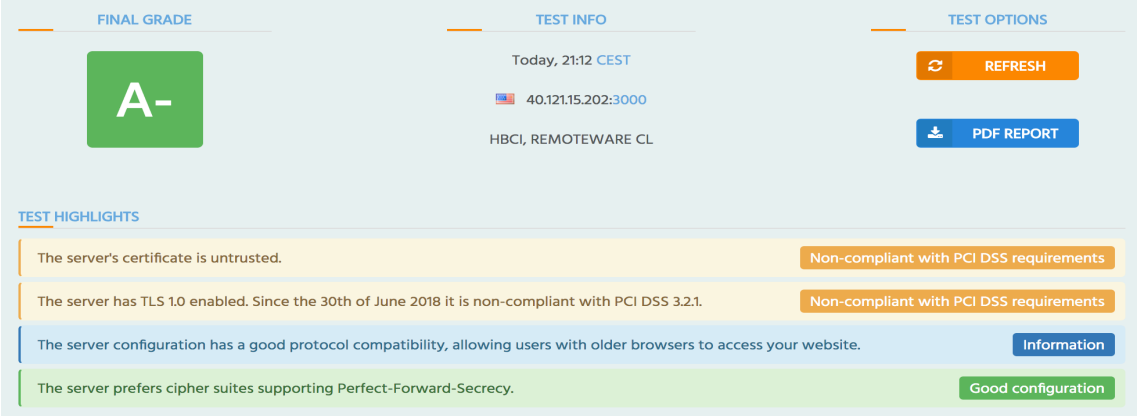
Las Figuras 5.7, 5.8 y 5.9 muestran más información sobre el análisis realizado por otras herramientas poniendo énfasis en la configuración de seguridad. Algunos datos informativos obtenidos son:

- El servidor presenta una buena configuración de seguridad con una calificación final de A- gracias al uso de las buenas prácticas recomendadas por la empresa Qualys SSL Labs[17].

[17]Para obtener más información sobre estas prácticas puede consultar la referencia [71].

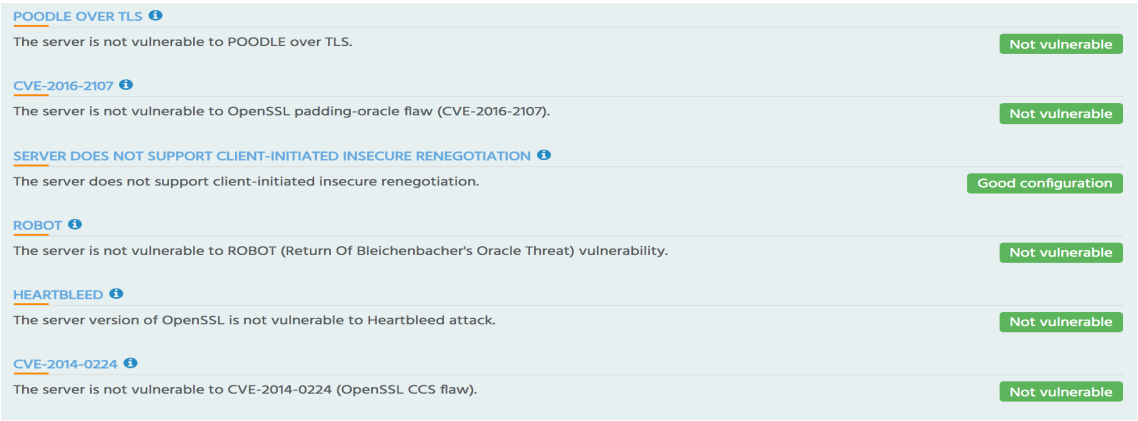
- Solamente se admiten conexiones a través del protocolo TLS (*Transport Layer Security*). Esta configuración es la idónea puesto que el protocolo SSL se considera inseguro debido a la gran cantidad de vulnerabilidades existentes: DROWN, FREAK, POODLE, etc.
- Se mitigan también otro tipo de vulnerabilidades: CRIME, Heartbleed, SLOTH, Robot, CVE-2014-0224 (OpenSSL CCS flaw), CVE-2016-2107 (OpenSSL padding-oracle flaw), etc.
- Solamente se permite la renegociación segura de los detalles de la comunicación. La renegociación insegura iniciada por el cliente no es soportada.

Figura 5.7: Análisis de seguridad con High-Tech Bridge - Resumen.



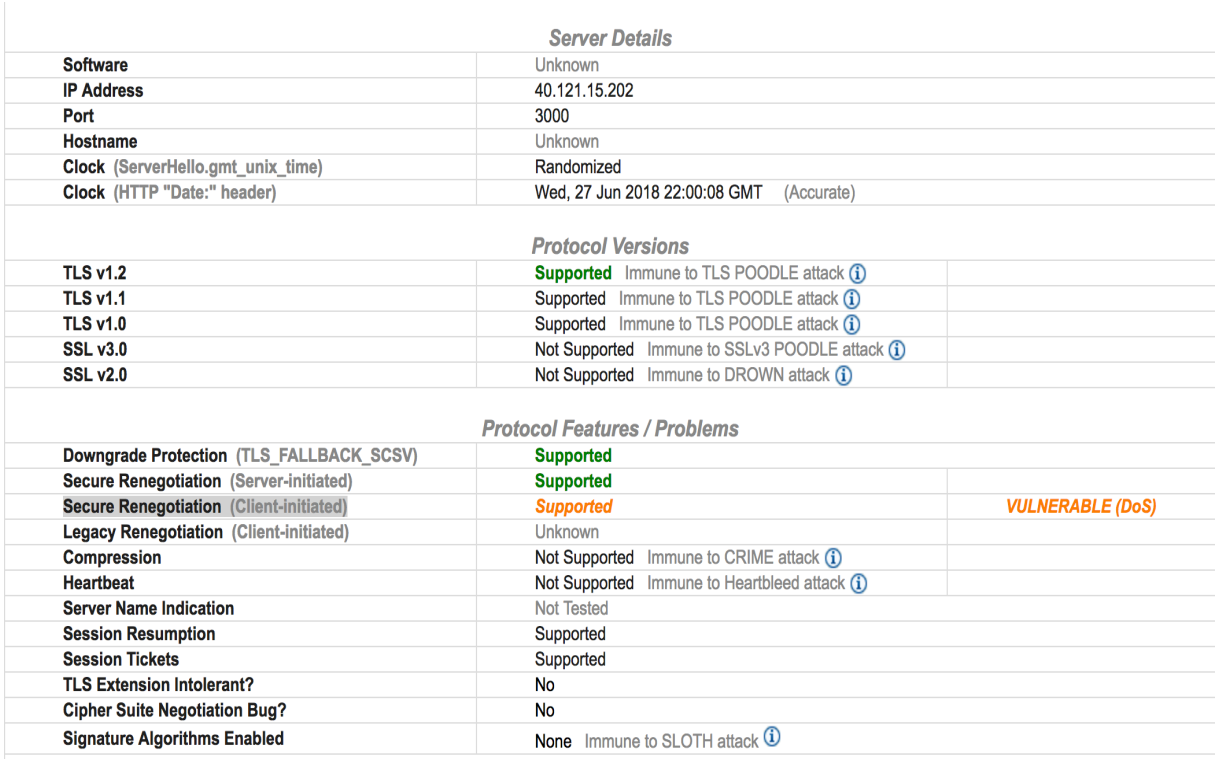
Fuente de comprobación: `https://www.htbridge.com/ssl/`

Figura 5.8: Análisis de seguridad con High-Tech Bridge - Vulnerabilidades no soportadas.



Fuente de comprobación: `https://www.htbridge.com/ssl/`

Figura 5.9: Análisis de seguridad con COMODO SSL Analyzer.



Server Details		
Software	Unknown	
IP Address	40.121.15.202	
Port	3000	
Hostname	Unknown	
Clock (ServerHello.gmt_unix_time)	Randomized	
Clock (HTTP "Date:" header)	Wed, 27 Jun 2018 22:00:08 GMT (Accurate)	
Protocol Versions		
TLS v1.2	Supported Immune to TLS POODLE attack	
TLS v1.1	Supported Immune to TLS POODLE attack	
TLS v1.0	Supported Immune to TLS POODLE attack	
SSL v3.0	Not Supported Immune to SSLv3 POODLE attack	
SSL v2.0	Not Supported Immune to DROWN attack	
Protocol Features / Problems		
Downgrade Protection (TLS_FALLBACK_SCSV)	Supported	
Secure Renegotiation (Server-initiated)	Supported	
Secure Renegotiation (Client-initiated)	Supported	VULNERABLE (DoS)
Legacy Renegotiation (Client-initiated)	Unknown	
Compression	Not Supported Immune to CRIME attack	
Heartbeat	Not Supported Immune to Heartbleed attack	
Server Name Indication	Not Tested	
Session Resumption	Supported	
Session Tickets	Supported	
TLS Extension Intolerant?	No	
Cipher Suite Negotiation Bug?	No	
Signature Algorithms Enabled	None Immune to SLOTH attack	

Fuente de comprobación: `https://sslanalyzer.comodoca.com/`

Por último, se muestra un ejemplo de la configuración negociada entre el cliente y el servidor para el navegador Google Chrome v67.0.3396.99 (Figura 5.10).

Figura 5.10: Configuración negociada en el navegador Google Chrome.

Capítulo 6

Casos de prueba

Toda implementación *software* requiere de varias fases de pruebas previas a la liberación de la versión de producción con el fin de garantizar la calidad en términos funcionales, de diseño, de seguridad, de rendimiento, etc. La información obtenida de este proceso es de vital importancia debido a que brinda la posibilidad de una detección temprana de errores e implicará en gran medida en el éxito o no del proyecto.

6.1. Introducción a los casos de prueba

Las pruebas o *testing* son un conjunto de prácticas ubicadas en el proceso de control de calidad que se definen como el procedimiento de investigación y análisis de la calidad del *software* con el fin principal de detectar errores de forma temprana y de proveer información útil a todos los interesados del proyecto o *stakeholders*. Dentro del conjunto de pruebas, éstas se pueden diferenciar en función de los aspectos a ser probados:

- Funcionalidad, donde se prueba que el sistema funcione como se espera y para lo que fue diseñado. Ejemplo de pruebas de esta categoría: pruebas unitarias, de integración, de regresión, de aceptación, etc.
- Rendimiento, donde se localizan las pruebas de carga y de estrés cuyo objetivo es conocer el comportamiento del sistema ante diferentes situaciones de saturación y el punto donde el sistema deja de funcionar correctamente.
- Usabilidad y adaptabilidad para conocer si el sistema es intuitivo, fácil de usar y se adapta a diferentes entornos y tipos de personas. Ejemplo de pruebas de esta categoría: pruebas moderadas, *card sorting*, pruebas A/B, evaluaciones heurísticas, etc.
- Seguridad para evaluar cómo de seguro es el sistema ante amenazas externas y para detectar vulnerabilidades. Ejemplo de pruebas de esta categoría: pruebas de penetración (*pentesting*).

El marco temporal donde se debe ejecutar un determinado tipo de prueba varía en función de sus objetivos. Algunos se pueden aplicar desde fases tempranas del desarrollo como es el caso de las pruebas unitarias -verifican el correcto funcionamiento de los componentes del sistema- o las pruebas de integración -verifican la correcta interacción entre los distintos componentes- mientras que otros requieren de un producto mínimo viable (*Minimum Viable Product* -MVP-) para poder obtener una retroalimentación como el caso de las pruebas *End-to-End*, pruebas automatizadas del flujo completo del sistema que simulan al usuario final en un escenario de producción interactuando con la interfaz de usuario (*User Interface* -UI-).

En Hyot, durante la fase de construcción y transición a medida que se iba implementando la funcionalidad se han ido ejecutando pruebas unitarias para verificar el correcto funcionamiento de cada caso de uso involucrado en cada iteración y pruebas de integración para constatar la interacción entre las diferentes partes que componen este proyecto con el fin de garantizar la calidad del sistema y cumplir con los requisitos tanto funcionales como no funcionales que se han ido marcando en cada iteración.

6.2. Casos de prueba

En esta sección se detallan los casos de prueba[1] identificados y a los que se ha sometido cada componente del proyecto Hyot, así como los resultados de los mismos. Para describir cada caso de prueba se ha utilizado el siguiente esquema:

- **Caso de prueba:** identificador del caso de prueba.
- **Descripción:** descripción de la funcionalidad a probar.
- **Salida esperada:** resultado esperado tras la finalización de la prueba.
- **Salida obtenida:** resultado obtenido tras la finalización de la prueba.
- **Resultado de la prueba:** resultado final de la prueba.

6.2.1. Componente - Configuración del dispositivo Raspberry Pi

Los casos de prueba a continuación descritos hacen referencia al componente del proyecto Hyot que permite la configuración inicial del dispositivo Raspberry Pi (RPi)[2].

[1] Únicamente se plasman aquellos casos de prueba considerados de mayor importancia para no dilatar demasiado el documento. En algunos casos, el caso de prueba -marcado con el símbolo *- puede aplicar a varios componentes aunque puede diferir la salida exacta mostrada.

[2] Aquellos casos de prueba marcados con el símbolo ** hacen referencia a la instalación de dependencias donde solamente se especificarán los referidos a paquetes, omitiendo aquellos referidos a librerías Python por su similitud.

Caso de prueba	UC-0001
Descripción	**Ejecución sin el fichero de utilidades utils.sh**
Salida esperada	Mensaje informativo y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a la no localización del fichero
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.1: Caso de prueba - Ejecución sin el fichero de utilidades utils.sh

Caso de prueba	UC-0002
Descripción	**Ejecución con usuario normal: pi ***
Salida esperada	Mensaje informativo: "*This component must be run as root*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a que el usuario no es un usuario superprivilegiado
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.2: Caso de prueba - Ejecución con usuario normal: pi

Caso de prueba	UC-0003
Descripción	**Ejecución con sudo y usuario normal: pi ***
Salida esperada	Continuación de la ejecución
Salida obtenida	Comprobación de usuario correcta y continuación de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.3: Caso de prueba - Ejecución con sudo y usuario normal: pi

Caso de prueba	UC-0004
Descripción	**Ejecución con usuario superprivilegiado: root ***
Salida esperada	Continuación de la ejecución
Salida obtenida	Comprobación de usuario correcta y continuación de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.4: Caso de prueba - Ejecución con usuario superprivilegiado: root

Caso de prueba	UC-0005
Descripción	**Ejecución en el sistema operativo (SO): Windows 10 ***
Salida esperada	Mensaje informativo: "*This component must be run on GNU/Linux platform (e.g. Raspbian)*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución al ser una plataforma no permitida
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.5: Caso de prueba - Ejecución en el sistema operativo (SO): Windows 10

Caso de prueba	UC-0006
Descripción	**Ejecución en el SO: MacOS (*Macintosh Operating System*) High Sierra ***
Salida esperada	Mensaje informativo: "*This component must be run on GNU/Linux platform (e.g. Raspbian)*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución al ser una plataforma no permitida
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.6: Caso de prueba - Ejecución en el SO: MacOS High Sierra

Caso de prueba	UC-0007
Descripción	**Ejecución en el SO: Ubuntu ***
Salida esperada	Mensaje informativo: "*This component must be run on a Raspberry Pi*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución al ser una plataforma GNU/Linux no permitida
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.7: Caso de prueba - Ejecución en el SO: Ubuntu

Caso de prueba	UC-0008
Descripción	**Ejecución en el SBC (*Single Board Computer*): Arduino ***
Salida esperada	Mensaje informativo: "*This component must be run on a Raspberry Pi*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución al ser un dispositivo no permitido
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.8: Caso de prueba - Ejecución en el SBC (Single Board Computer): Arduino

Caso de prueba	UC-0009
Descripción	**Ejecución en el SBC: RPi 2 ***
Salida esperada	Continuación de la ejecución
Salida obtenida	Comprobación de plataforma y dispositivo correcta. Continuación de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.9: Caso de prueba - Ejecución en el SBC: RPi 2

Caso de prueba	UC-0010
Descripción	**Ejecución sin el comando wget instalado**
Salida esperada	Mensaje informativo: "*Command not found: wget. Please, install this command to check if the network connection is available*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a que el comando no se encuentra instalado en la RPi
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.10: Caso de prueba - Ejecución sin el comando wget instalado

Caso de prueba	UC-0011
Descripción	**Ejecución sin conexión a Internet** *
Salida esperada	Mensaje informativo: "*Raspberry Pi is not connected to the network. Please, enable the network to continue the setup*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a que la RPi no posee red
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.11: Caso de prueba - Ejecución sin conexión a Internet

Caso de prueba	UC-0012
Descripción	**Ejecución sin el comando pgrep instalado**
Salida esperada	Mensaje informativo: "*Command not found: pgrep. Please, install this command to check and avoid the concurrency*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a que el comando no se encuentra instalado en la RPi
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.12: Caso de prueba - Ejecución sin el comando pgrep instalado

Caso de prueba	UC-0013
Descripción	**Ejecución cuando otra instancia ya está ejecutándose** *
Salida esperada	Mensaje informativo: "*Process: raspberrypi_setup.sh is already running with PID [pid]*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a que otra instancia está en ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.13: Caso de prueba - Ejecución cuando otra instancia ya está ejecutándose

Caso de prueba	UC-0014
Descripción	**Ejecución con las opciones: -h, -v y -p**
Salida esperada	Mensaje informativo: "*Invalid parameter number. Please, type the '-h' or '–help' option to show the help*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a que el número de opciones introducido es inválido
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.14: Caso de prueba - Ejecución con las opciones: –h, -v y -p

Caso de prueba	UC-0015
Descripción	**Ejecución con la opción: -h o –help ***
Salida esperada	Se muestra la ayuda
Salida obtenida	La ayuda es mostrada y la ejecución es finalizada posteriormente
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.15: Caso de prueba - Ejecución con la opción: -h o –help

Caso de prueba	UC-0016
Descripción	**Ejecución con la opción: -v o –verbose**
Salida esperada	Se activa el modo *verbose*
Salida obtenida	La ejecución continúa y se muestra más información sobre ésta
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.16: Caso de prueba - Ejecución con la opción: -v o –verbose

Caso de prueba	UC-0017
Descripción	**Ejecución con la opción: -p o –packages**
Salida esperada	Ejecuta solamente la instalación de paquetes y librerías Python
Salida obtenida	Instala y/o actualiza las dependencias
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.17: Caso de prueba - Ejecución con la opción: -p o –packages

Caso de prueba	UC-0018
Descripción	**Ejecución con la opción: -i o –interfaces**
Salida esperada	Ejecuta solamente la activación de las interfaces
Salida obtenida	Habilita las interfaces
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.18: Caso de prueba - Ejecución con la opción: -i o –interfaces

Caso de prueba	UC-0019
Descripción	**Ejecución con las opciones: -v y -p**
Salida esperada	Ejecuta solamente la instalación de paquetes y librerías Python activando el modo *verbose*
Salida obtenida	Instala y/o actualiza las dependencias mostrando más información sobre el proceso
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.19: Caso de prueba - Ejecución con las opciones: -v y -p

Caso de prueba	UC-0020
Descripción	**Ejecución con las opciones: -v y -i**
Salida esperada	Ejecuta solamente la activación de interfaces habilitando el modo *verbose*
Salida obtenida	Habilita las interfaces mostrando más información sobre el proceso
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.20: Caso de prueba - Ejecución con las opciones: -v y -i

Caso de prueba	UC-0021
Descripción	**Ejecución con una opción inválida (-a) ***
Salida esperada	Mensaje informativo: "*Unknown option: -a. Please, type the option '-h' or '–help' to show the help*" y finalización de la ejecución
Salida obtenida	Mensaje y finalización de la ejecución debido a una opción inválida
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.21: Caso de prueba - Ejecución con una opción inválida (-a)

Caso de prueba	UC-0022
Descripción	**Ejecución sin los comandos apt-get, apt-cache y/o dpkg instalados**
Salida esperada	Mensaje informativo: "*Command line tool: [command] is not installed in the system. Please, install this package before continuing*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a que alguno o todos los comandos no se encuentran instalados en la RPi
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.22: Caso de prueba - Ejecución sin los comandos apt-get, apt-cache y/o dpkg instalados

Caso de prueba	UC-0023
Descripción	**Ejecución en un entorno sin el paquete python-pip instalado ****
Salida esperada	Mensaje informativo: "*Package: python-pip was installed successfully*" y continuación de la ejecución
Salida obtenida	Instalación del paquete y continuación de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.23: Caso de prueba - Ejecución en un entorno sin el paquete python-pip instalado

Caso de prueba	UC-0024
Descripción	**Ejecución en un entorno sin el paquete python-pip instalado. Error de instalación ****
Salida esperada	Mensaje informativo: "*Error to install the package: python-pip*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a un error
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.24: Caso de prueba - Ejecución en un entorno sin el paquete python-pip instalado. Error de instalación

Caso de prueba	UC-0025
Descripción	**Ejecución en un entorno sin el paquete gnupg-nonexistent instalado ****
Salida esperada	Mensaje informativo: "*Package: gnupg-nonexistent not found in the repository. Please, check its name*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a que el paquete no existe en el repositorio
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.25: Caso de prueba - Ejecución en un entorno sin el paquete gnupg-nonexistent instalado

Caso de prueba	UC-0026
Descripción	**Ejecución en un entorno con el paquete i2c-tools instalado y actualizado ****
Salida esperada	Mensaje informativo: "*Package is already updated to the last version*" y continuación de la ejecución
Salida obtenida	Mensaje informativo y continuación de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.26: Caso de prueba - Ejecución en un entorno con el paquete i2c-tools instalado y actualizado

Caso de prueba	UC-0027
Descripción	**Ejecución en un entorno con el paquete i2c-tools instalado pero no actualizado ****
Salida esperada	Mensaje informativo: "*Package: i2c-tools is installed and updated in the system*" y continuación de la ejecución
Salida obtenida	Actualización del paquete y continuación de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.27: Caso de prueba - Ejecución en un entorno con el paquete i2c-tools instalado pero no actualizado

Caso de prueba	UC-0028
Descripción	**Ejecución en un entorno con el paquete i2c-tools instalado pero no actualizado. Error de actualización ****
Salida esperada	Mensaje informativo: "*Error to update the package: i2c-tools*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a un error
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.28: Caso de prueba - Ejecución en un entorno con el paquete i2c-tools instalado pero no actualizado. Error de actualización

Caso de prueba	UC-0029
Descripción	**Ejecución sin el comando raspi-config instalado**
Salida esperada	Mensaje informativo: "*Command not found: raspi-config. Please, run this component on a Raspberry Pi with Raspbian platform*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a que el comando no se encuentra instalado
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.29: Caso de prueba - Ejecución sin el comando raspi-config instalado

Caso de prueba	UC-0030
Descripción	**Ejecución con interfaces definidas correctamente**
Salida esperada	Mensaje informativo: "*Interface enabled: i2c/camera*" y habilitación de las interfaces
Salida obtenida	Interfaces habilitadas en la RPi y continuación de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.30: Caso de prueba - Ejecución con interfaces definidas correctamente

Caso de prueba	UC-0031
Descripción	**Ejecución con interfaz definida erróneamente (i2c-nonexistent)**
Salida esperada	Mensaje informativo: "*Error to enable the interface: i2c-nonexistent*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a que la interfaz no existe
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.31: Caso de prueba - Ejecución con interfaz definida erróneamente (i2c-nonexistent)

Caso de prueba	UC-0032
Descripción	**Ejecución sin el comando reboot instalado**
Salida esperada	Mensaje informativo: "*Command not found: reboot. Please, reboot the system manually*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a que el comando no se encuentra instalado en la RPi
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.32: Caso de prueba - Ejecución sin el comando reboot instalado

Caso de prueba	UC-0033
Descripción	**El usuario introduce el carácter 'y' a la pregunta de si desea reiniciar el sistema tras finalizar la ejecución**
Salida esperada	Mensaje informativo: "*Rebooting the system...*" y reinicio de la RPi
Salida obtenida	Mensaje informativo y reinicio de la RPi después de 5 segundos
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.33: Caso de prueba - El usuario introduce el carácter 'y' a la pregunta de si desea reiniciar el sistema tras finalizar la ejecución

Caso de prueba	UC-0034
Descripción	**El usuario introduce el carácter 'n' a la pregunta de si desea reiniciar el sistema tras finalizar la ejecución**
Salida esperada	Finalización de la ejecución
Salida obtenida	Finalización de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.34: Caso de prueba - El usuario introduce el carácter 'n' a la pregunta de si desea reiniciar el sistema tras finalizar la ejecución

Caso de prueba	UC-0035
Descripción	**Pulsación de Control+C durante la ejecución ***
Salida esperada	Mensaje informativo: "*Exception: KeyboardInterrupt. Please, wait until the process finishes*" y finalización de la ejecución
Salida obtenida	Mensaje informativo sobre la interrupción de la ejecución y finalización de ésta
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.35: Caso de prueba - Pulsación de Control+C durante la ejecución

6.2.2. Componente - Monitorización de sucesos del entorno

Los casos de prueba a continuación descritos hacen referencia al componente del proyecto Hyot que permite monitorizar sucesos del entorno y actuar ante lecturas anómalas con el fin de generar una prueba veraz e irrefutable de la incidencia producida.

Caso de prueba	UC-0036
Descripción	**Ejecución sin un módulo Python instalado ***
Salida esperada	Mensaje informativo: "*Error to import in hyot_main: no module named [module]*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a que el módulo Python no se encuentra instalado en la RPi
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.36: Caso de prueba - Ejecución sin un módulo Python instalado

Caso de prueba	UC-0037
Descripción	**Ejecución sin indicar un valor asociado a una opción ***
Salida esperada	Mensaje informativo: "*hyot_main.py: error: argument [option]: expected one argument*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a la no indicación de un valor para alguna opción introducida
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.37: Caso de prueba - Ejecución sin indicar un valor asociado a una opción

Caso de prueba	UC-0038
Descripción	**Ejecución indicando un valor inválido a una opción**
Salida esperada	Mensaje informativo y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a la indicación de un valor no permitido, ya sea por su tipo o porque se encuentra fuera de un rango específico
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.38: Caso de prueba - Ejecución indicando un valor inválido a una opción

Caso de prueba	UC-0039
Descripción	**Ejecución sin introducir ninguna opción**
Salida esperada	Opciones con valor por defecto y continuación de la ejecución
Salida obtenida	Uso de valores por defecto para las opciones y continuación de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.39: Caso de prueba - Ejecución sin introducir ninguna opción

Caso de prueba	UC-0040
Descripción	**Ejecución introduciendo valores válidos en las opciones**
Salida esperada	Ejecución con los valores especificados
Salida obtenida	Uso de valores definidos para las opciones en lugar de los establecidos por defecto y continuación de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.40: Caso de prueba - Ejecución introduciendo valores válidos en las opciones

Caso de prueba	UC-0041
Descripción	**Ejecución indicando un valor válido en las opciones relacionadas a componentes electrónicos pero difiriendo de la conexión física**
Salida esperada	Continuación de la ejecución pero con un funcionamiento incorrecto
Salida obtenida	Funcionamiento incorrecto al no poder establecer comunicación con los componentes electrónicos
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.41: Caso de prueba - Ejecución indicando un valor válido en las opciones relacionadas a componentes electrónicos pero difiriendo de la conexión física

Caso de prueba	UC-0042
Descripción	**Ejecución sin conectar algún componente electrónico al prototipo**
Salida esperada	Mensaje informativo y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución al no poder establecer comunicación con el componente
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.42: Caso de prueba - Ejecución sin conectar algún componente electrónico al prototipo

Caso de prueba	UC-0043
Descripción	**Desconectar algún componente electrónico del prototipo una vez ejecutado**
Salida esperada	Mensaje informativo y finalización o continuación de la ejecución en función del componente
Salida obtenida	Mensaje informativo y finalización de la ejecución al no poder establecer comunicación con el componente o no obtención de la medición y continuación de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.43: Caso de prueba - Desconectar algún componente electrónico del prototipo una vez ejecutado

Caso de prueba	UC-0044
Descripción	**Inicialización de servicios y utilidades con datos por defecto**
Salida esperada	Inicialización correcta y continuación de la ejecución
Salida obtenida	Mensajes informativos sobre la inicialización correcta de todos los servicios y utilidades incluyendo la generación o apertura de base de datos (BBDD), directorios, etc. y continuación de la ejecución con la monitorización de sucesos del entorno
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.44: Caso de prueba - Inicialización de servicios y utilidades con datos por defecto

Caso de prueba	UC-0045
Descripción	**Inicialización de servicios y utilidades con datos correctos**
Salida esperada	Inicialización correcta y continuación de la ejecución
Salida obtenida	Mensajes informativos sobre la inicialización correcta de todos los servicios y utilidades incluyendo la generación o apertura de BBDD, directorios, etc. y continuación de la ejecución con la monitorización de sucesos del entorno
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.45: Caso de prueba - Inicialización de servicios y utilidades con datos correctos

Caso de prueba	UC-0046
Descripción	**Inicialización de servicios y utilidades con datos correctos. Error durante la inicialización**
Salida esperada	Mensaje informativo: "*Error to initialize the [name] module. Exception:...*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución al producirse un error durante la inicialización de algún servicio o utilidad
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.46: Caso de prueba - Inicialización de servicios y utilidades con datos correctos. Error durante la inicialización

Caso de prueba	UC-0047
Descripción	**Inicialización de servicios y utilidades con datos incorrectos**
Salida esperada	Mensaje informativo y finalización de la ejecución
Salida obtenida	Error en la inicialización de algún servicio o utilidad (p.ej. dato introducido vacío, credenciales o token erróneo, servidor no corresponde a una red de negocio, etc.) y finalización de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.47: Caso de prueba - Inicialización de servicios y utilidades con datos incorrectos

Caso de prueba	UC-0048
Descripción	**Generar par de claves**
Salida esperada	Generación del par de claves y continuación de la ejecución
Salida obtenida	Generación de la clave pública y privada a utilizar, del código QR asociado y exportación a un fichero `.asc`. Continuación de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.48: Caso de prueba - Generar par de claves

Caso de prueba	UC-0049
Descripción	**Seleccionar par de claves a utilizar**
Salida esperada	Par de claves seleccionado y continuación de la ejecución
Salida obtenida	Selección de la clave pública y privada a utilizar entre las existentes indicando su *fingerprint* y continuación de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.49: Caso de prueba - Seleccionar par de claves a utilizar

Caso de prueba	UC-0050
Descripción	**Seleccionar varios pares de claves a utilizar**
Salida esperada	Pares de claves seleccionados y continuación de la ejecución
Salida obtenida	Selección de varios pares de claves entre las existentes para la encriptación y solamente un par de claves para el firmado. Continuación de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.50: Caso de prueba - Seleccionar varios pares de claves a utilizar

Caso de prueba	UC-0051
Descripción	**Introducir contraseña de la clave privada ***
Salida esperada	Comprobación de contraseña no vacía y continuación de la ejecución
Salida obtenida	Continuación de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.51: Caso de prueba - Introducir contraseña de la clave privada

Caso de prueba	UC-0052
Descripción	**Introducir contraseña vacía de la clave privada. Número de intentos no superado ***
Salida esperada	Mensaje informativo: "*The password can not be empty. Please, try it again*" y solicitud de nuevo de la contraseña
Salida obtenida	Solicitud de nuevo de introducción de contraseña
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.52: Caso de prueba - Introducir contraseña vacía de la clave privada. Número de intentos no superado

Caso de prueba	UC-0053
Descripción	**Introducir contraseña vacía de la clave privada. Número de intentos no superado ***
Salida esperada	Mensaje informativo: "*Number of attempts spent. Please, run again the code*" y finalización de la ejecución
Salida obtenida	Finalización de la ejecución al haber superado el número intentos límite
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.53: Caso de prueba - Introducir contraseña vacía de la clave privada. Número de intentos superado

Caso de prueba	UC-0054
Descripción	**Inicialización del servicio Dropbox. Espacio disponible insuficiente**
Salida esperada	Mensaje informativo: "*Warning! The available space may be insufficient (500 MB). It is advisable to increase it before continuing the execution...*" y continuación de la ejecución
Salida obtenida	Inicialización del servicio informando al usuario de que dispone menos de 500 MB de espacio lo cual puede ser insuficiente y continuación de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.54: Caso de prueba - Inicialización del servicio Dropbox. Espacio disponible insuficiente

Caso de prueba	UC-0055
Descripción	**Inicialización del servicio Hyperledger Fabric (HF). Servidor de Hyperledger Composer (HC) no contiene una red de negocio desplegada**
Salida esperada	Mensaje informativo: "*Response of Ping request is not a JSON serializable or it does not contain the participant key*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución al no encontrarse una Blockchain (BC) desplegada
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.55: Caso de prueba - Inicialización del servicio Hyperledger Fabric (HF) - Servidor de Hyperledger Composer (HC) no contiene una red de negocio desplegada

Caso de prueba	UC-0056
Descripción	**Securizar el servidor HC. Peticiones con *api-key***
Salida esperada	Comprobación de la petición y continuación de la ejecución
Salida obtenida	Petición exitosa al incorporar el *api-key* de seguridad y continuación de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.56: Caso de prueba - Securizar el servidor HC. Peticiones con api-key

Caso de prueba	UC-0057
Descripción	**Securizar el servidor HC. Peticiones sin *api-key***
Salida esperada	Mensaje informativo: "*Error 401: Unauthorized request. Please, enter a valid API key or credentials to submit the request to the Hyperledger Composer REST server*" y finalización de la ejecución
Salida obtenida	Petición denegada al no incorporar el *api-key* de seguridad y finalización de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.57: Caso de prueba - Securizar el servidor HC. Peticiones sin api-key

Caso de prueba	UC-0058
Descripción	**Monitorización de sucesos correctos**
Salida esperada	Comprobación de suceso normal y continuación de la ejecución
Salida obtenida	Acciones de un suceso normal son ejecutadas y continuación con la siguiente medición
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.58: Caso de prueba - Monitorización de sucesos correctos

Caso de prueba	UC-0059
Descripción	**Monitorización de suceso anómalos**
Salida esperada	Activación del protocolo de alerta y continuación de la ejecución
Salida obtenida	Acciones de un suceso anómalo son ejecutadas y continuación con la siguiente medición
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.59: Caso de prueba - Monitorización de suceso anómalos

Caso de prueba	UC-0060
Descripción	**Monitorización de suceso anómalos. Notificación vía *email* activada**
Salida esperada	Activación del protocolo de alerta, envío de notificación y continuación de la ejecución
Salida obtenida	Acciones de un suceso anómalo son ejecutadas incluyendo el envío de una notificación a la dirección de correo especificada y continuación con la siguiente medición
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.60: Caso de prueba - Monitorización de suceso anómalos. Notificación vía email activada

Caso de prueba	UC-0061
Descripción	**Error durante una medición**
Salida esperada	Mensaje informativo, envío de notificación en caso de estar activada la opción y finalización de la ejecución
Salida obtenida	Finalización de la ejecución notificando a la dirección de correo en caso de estar especificada y adjuntando el fichero *log*. Finalización de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.61: Caso de prueba - Error durante una medición

Caso de prueba	UC-0062
Descripción	**Medición con valores incorrectos obtenidos**
Salida esperada	Comprobación de valores y continuación de la ejecución
Salida obtenida	Continuación con la siguiente medición al obtener valores vacíos o extraños los cuales son ignorados
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.62: Caso de prueba - Medición con valores incorrectos obtenidos

Caso de prueba	UC-0063
Descripción	**Finalización ordenada de los servicios y utilidades**
Salida esperada	Finalización ordenada de la ejecución
Salida obtenida	Al enviar una señal de terminación, se finaliza la ejecución de manera ordenada destruyendo las instancias inicializadas
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.63: Caso de prueba - Finalización ordenada de los servicios y utilidades

6.2.3. Componente - Desencriptación de evidencia

Los casos de prueba a continuación descritos hacen referencia al componente del proyecto Hyot que permite la desencriptación de evidencias previamente encriptadas y firmadas con la herramienta GPG, la verificación de la firma e integridad del contenido.

Caso de prueba	UC-0064
Descripción	**Ejecución sin las opciones obligatorias (-g/–gpghome y -ha/–hash)**
Salida esperada	Mensaje informativo: "*hyot_decryption.py: error: argument [option] is required*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a la no indicación de una opción obligatoria
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.64: Caso de prueba - Ejecución sin las opciones obligatorias (-g/–gpghome y -ha/–hash)

Caso de prueba	UC-0065
Descripción	**Ejecución sin introducir un método para indicar la evidencia a usar**
Salida esperada	Mensaje informativo: "*Please, enter some method to indicate the evidence to use or type the -h/–help option to get more information*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a la ausencia de un método para indicar la evidencia a utilizar
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.65: Caso de prueba - Ejecución sin introducir un método para indicar la evidencia a usar

Caso de prueba	UC-0066
Descripción	**Ejecución indicando dos métodos para la evidencia a utilizar (-e/–encryptedfile y -l/–link)**
Salida esperada	Mensaje informativo: "*Please, enter only one way to indicate the evidence to use (local file or link) or type the -h/–help option to get more information*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a la introducción de dos métodos para indicar la evidencia a utilizar
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.66: Caso de prueba - Ejecución indicando dos métodos para la evidencia a utilizar (-e/–encryptedfile y -l/–link)

Caso de prueba	UC-0067
Descripción	**Ejecución sin introducir un método para indicar el par de claves a usar**
Salida esperada	Mensaje informativo: "*Please, enter some method to indicate the pair of keys to use or type the -h/–help...*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a la ausencia de un método para indicar el par de claves a utilizar
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.67: Caso de prueba - Ejecución sin introducir un método para indicar el par de claves a usar

Caso de prueba	UC-0068
Descripción	**Ejecución indicando dos métodos para el par de claves a utilizar (-f/–fingerprint y -k/–keys)**
Salida esperada	Mensaje informativo: "*Please, enter only one method (by means of fingerprint or file) to indicate the pair of keys to use or type the -h/–help option to get more information*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a la introducción de dos métodos para indicar el par de claves a utilizar
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.68: Caso de prueba - Ejecución indicando dos métodos para el par de claves a utilizar (-f/–fingerprint y -k/–keys)

Caso de prueba	UC-0069
Descripción	**Ejecución indicando el enlace para la evidencia a utilizar (opción -l/–link)**
Salida esperada	Descarga de la evidencia y continuación de la ejecución
Salida obtenida	Verificación de conexión a Internet, del formato del enlace introducido y descarga de la evidencia encriptada y firmada al sistema local
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.69: Caso de prueba - Ejecución indicando el enlace para la evidencia a utilizar (opción -l/–link)

Caso de prueba	UC-0070
Descripción	**Ejecución indicando en todas las opciones directorios o ficheros existentes**
Salida esperada	Comprobación de existencia correcta y continuación de la ejecución
Salida obtenida	Continuación de la ejecución debido a que todos los directorios y ficheros indicados existen en el sistema local
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.70: Caso de prueba - Ejecución indicando en todas las opciones directorios o ficheros existentes

Caso de prueba	UC-0071
Descripción	**Ejecución indicando en alguna opción un directorio o fichero inexistente**
Salida esperada	Mensaje informativo y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a que el directorio o el fichero indicado no existe o la opción no permite el tipo indicado (p.ej. espera un directorio y se especificó un fichero existente o viceversa)
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.71: Caso de prueba - Ejecución indicando en alguna opción un directorio o fichero inexistente

Caso de prueba	UC-0072
Descripción	**Ejecución indicando una evidencia con formato .gpg**
Salida esperada	Comprobación de extensión de la evidencia correcta y continuación de la ejecución
Salida obtenida	Continuación de la ejecución debido a que la evidencia encriptada y firmada posee el formato adecuado
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.72: Caso de prueba - Ejecución indicando una evidencia con formato .gpg

Caso de prueba	UC-0073
Descripción	**Ejecución indicando una evidencia con formato .h264**
Salida esperada	Mensaje informativo: "*The encrypted and signed evidence has an extension which is not allowed. It must be a file with format: .gpg.*" y finalización de la ejecución
Salida obtenida	Mensaje informativo y finalización de la ejecución debido a que la evidencia encriptada y firmada no posee el formato adecuado
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.73: Caso de prueba - Ejecución indicando una evidencia con formato .h264

Caso de prueba	UC-0074
Descripción	**Ejecución indicando un fichero que contiene el par de claves**
Salida esperada	Comprobación del fichero correcta y continuación de la ejecución
Salida obtenida	La clave pública y privada son importadas correctamente y continuación de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.74: Caso de prueba - Ejecución indicando un fichero que contiene el par de claves

Caso de prueba	UC-0075
Descripción	**Ejecución indicando un fichero que contiene solamente la clave pública**
Salida esperada	Mensaje informativo: "*The entered key file does not contain the pair of keys (public and private key). Please, use the file generated in the component of monitoring of environmental events*" y finalización de la ejecución
Salida obtenida	Comprobación del fichero y finalización de la ejecución porque no contiene el par de claves
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.75: Caso de prueba - Ejecución indicando un fichero que contiene solamente la clave pública

Caso de prueba	UC-0076
Descripción	**Ejecución indicando el *fingerprint* asociado a un par de claves existente en el directorio GPG indicado**
Salida esperada	Comprobación del *fingerprint* y continuación de la ejecución
Salida obtenida	Selección del par de claves a utilizar y continuación de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.76: Caso de prueba - Ejecución indicando el fingerprint asociado a un par de claves existente en el directorio GPG indicado

Caso de prueba	UC-0077
Descripción	**Ejecución indicando un *fingerprint* no asociado a ningún par de claves existente en el directorio GPG indicado**
Salida esperada	Mensaje informativo: "*The entered fingerprint does not exist in the indicated GPG directory. Please, import the pair of keys to continue the process or use an existing fingerprint*" y finalización de la ejecución
Salida obtenida	Comprobación del *fingerprint* y finalización de la ejecución porque no está asociado a ningún par de claves existente
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.77: Caso de prueba - Ejecución indicando un fingerprint no asociado a ningún par de claves existente en el directorio GPG indicado

Caso de prueba	UC-0078
Descripción	**Ejecución indicando el *fingerprint* en un directorio GPG vacío**
Salida esperada	Mensaje informativo: "*The GPG directory does not contain any public or private key. Please, import the pair of keys with the -k/–keys option to continue the process*" y finalización de la ejecución
Salida obtenida	Comprobación del directorio GPG y finalización de la ejecución porque no contiene ningún par de claves
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.78: Caso de prueba - Ejecución indicando el fingerprint en un directorio GPG vacío

Caso de prueba	UC-0079
Descripción	**Ejecución indicando una contraseña de la clave privada correcta**
Salida esperada	Mensaje informativo: "*Evidence successfully decrypted in the path: [path]*" y continuación de la ejecución
Salida obtenida	Desencriptación de la evidencia y continuación de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.79: Caso de prueba - Ejecución indicando una contraseña de la clave privada correcta

Caso de prueba	UC-0080
Descripción	**Ejecución indicando una contraseña de la clave privada incorrecta**
Salida esperada	Mensaje informativo: "*The decryption has failed...*" y finalización de la ejecución
Salida obtenida	Finalización de la ejecución al producirse un error durante la desencriptación
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.80: Caso de prueba - Ejecución indicando una contraseña de la clave privada incorrecta

Caso de prueba	UC-0081
Descripción	**Ejecución indicando un par de claves diferente al utilizado durante la encriptación y firmado**
Salida esperada	Mensaje informativo: "*The decryption has failed...*" y finalización de la ejecución
Salida obtenida	Finalización de la ejecución al producirse un error durante la desencriptación
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.81: Caso de prueba - Ejecución indicando un par de claves diferente al utilizado durante la encriptación y firmado

Caso de prueba	UC-0082
Descripción	**Ejecución indicando una evidencia encriptada y firmada**
Salida esperada	Mensaje informativo: "*Information of the signature...*" y continuación de la ejecución
Salida obtenida	Evidencia es desencriptada (almacenada en el mismo directorio que la evidencia encriptada y firmada), firma es verificada y continuación de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.82: Caso de prueba - Ejecución indicando una evidencia encriptada y firmada

Caso de prueba	UC-0083
Descripción	**Ejecución indicando una evidencia encriptada y no firmada**
Salida esperada	Mensaje informativo: "*Evidence was not signed*" y continuación de la ejecución
Salida obtenida	Evidencia es desencriptada y continuación de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.83: Caso de prueba - Ejecución indicando una evidencia encriptada y no firmada

Caso de prueba	UC-0084
Descripción	**Ejecución indicando un valor *hash* correcto para la evidencia**
Salida esperada	Mensaje informativo: "*Both hash codes are the same. The evidence has not been altered and its integrity is guaranteed*" y finalización de la ejecución
Salida obtenida	Integridad del contenido es garantizado y finalización de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.84: Caso de prueba - Ejecución indicando un valor hash correcto para la evidencia

Caso de prueba	UC-0085
Descripción	**Ejecución indicando un valor *hash* incorrecto para la evidencia**
Salida esperada	Mensaje informativo: "*Both hash codes are different. The evidence may have been manipulated by a malicious third party and therefore its integrity is not guaranteed*" y finalización de la ejecución
Salida obtenida	Integridad del contenido no es garantizado y finalización de la ejecución
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.85: Caso de prueba - Ejecución indicando un valor hash incorrecto para la evidencia

Caso de prueba	UC-0086
Descripción	**Ejecución indicando un directorio de desencriptación (opción -d/–decryptedhome)**
Salida esperada	Evidencia desencriptada es almacenada en el directorio destino especificado
Salida obtenida	Proceso completo de desencriptación correcto y evidencia original es almacenada en el directorio destino
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.86: Caso de prueba - Ejecución indicando un directorio de desencriptación (opción -d/–decryptedhome)

6.2.4. Componente - Sistema web

Los casos de prueba a continuación descritos hacen referencia al componente del proyecto Hyot que permite la consulta en tiempo real de toda la información monitorizada del entorno.

Caso de prueba	UC-0087
Descripción	**Carga inicial de datos en el sistema web**
Salida esperada	El sistema web carga datos iniciales al arrancar
Salida obtenida	El sistema web contiene el usuario administrador establecido por defecto
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.87: Caso de prueba - Carga inicial de datos en el sistema web

Caso de prueba	UC-0088
Descripción	**Iniciar sesión con nombre de usuario o *email***
Salida esperada	Inicio de sesión correcto
Salida obtenida	El usuario inicia sesión correctamente en el sistema web
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.88: Caso de prueba - Iniciar sesión con nombre de usuario o email

Caso de prueba	UC-0089
Descripción	**Iniciar sesión con credenciales erróneas**
Salida esperada	Mensaje informativo notificando del error al usuario
Salida obtenida	El inicio de sesión no se efectúa debido a credenciales erróneas
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.89: Caso de prueba - Iniciar sesión con credenciales erróneas

Caso de prueba	UC-0090
Descripción	**Iniciar sesión con opción *Recordarme* activada**
Salida esperada	Inicio de sesión automático
Salida obtenida	La sesión se inicia automáticamente sin introducir las credenciales
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.90: Caso de prueba - Iniciar sesión con opción Recordarme activada

Caso de prueba	UC-0091
Descripción	**Iniciar sesión con usuario administrador**
Salida esperada	Inicio de sesión y redirección al panel de control
Salida obtenida	Se muestra el panel de control
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.91: Caso de prueba - Iniciar sesión con usuario administrador

Caso de prueba	UC-0092
Descripción	**Iniciar sesión con usuario normal**
Salida esperada	Inicio de sesión y redirección a la página de usuario
Salida obtenida	Se muestra la página de usuario
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.92: Caso de prueba - Iniciar sesión con usuario normal

Caso de prueba	UC-0093
Descripción	**Iniciar sesión con usuario sin rol**
Salida esperada	Inicio de sesión y redirección a una página informativa
Salida obtenida	Redirección a la página *no rol* para informar del problema al usuario
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.93: Caso de prueba - Iniciar sesión con usuario sin rol

Caso de prueba	UC-0094
Descripción	**Iniciar sesión con cuenta bloqueada, inactiva o expirada o contraseña expirada**
Salida esperada	Se muestra un mensaje de error al usuario
Salida obtenida	El usuario no puede iniciar sesión y es informado del error
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.94: Caso de prueba - Iniciar sesión con cuenta bloqueada, inactiva o expirada o contraseña expirada

Caso de prueba	UC-0095
Descripción	**Iniciar sesión con usuario normal existiendo otra sesión iniciada**
Salida esperada	El sistema web deniega el acceso informando al usuario del error
Salida obtenida	El usuario normal no puede iniciar sesión hasta que la anterior sesión sea invalidada puesto que presenta el límite de 1 sesión
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.95: Caso de prueba - Iniciar sesión con usuario normal existiendo otra sesión iniciada

Caso de prueba	UC-0096
Descripción	**Iniciar sesión con administrador existiendo otra sesión iniciada**
Salida esperada	El sistema web inicia sesión en el nuevo acceso
Salida obtenida	El usuario administrador inicia sesión en el nuevo dispositivo o navegador y la anterior sesión sigue siendo válida
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.96: Caso de prueba - Iniciar sesión con administrador existiendo otra sesión iniciada

Caso de prueba	UC-0097
Descripción	**Cerrar sesión**
Salida esperada	El usuario finaliza su sesión y no puede acceder al sistema web hasta que inicie sesión de nuevo introduciendo sus credenciales
Salida obtenida	La sesión de usuario se finaliza redireccionando a la página de inicio
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.97: Caso de prueba - Cerrar sesión

Caso de prueba	UC-0098
Descripción	**Restablecer contraseña**
Salida esperada	Envío de *email* a la dirección del usuario incluyendo un *token* de seguridad válido durante 30 minutos
Salida obtenida	El usuario recibe un *email* con las instrucciones a seguir
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.98: Caso de prueba - Restablecer contraseña

Caso de prueba	UC-0099
Descripción	**Indicar contraseña válida (p.ej. 8Kio5_ff)**
Salida esperada	El sistema web almacena la nueva contraseña del usuario
Salida obtenida	El usuario restaura su contraseña
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.99: Caso de prueba - Indicar contraseña válida (p.ej. 8Kio5_ff)

Caso de prueba	UC-0100
Descripción	**Indicar contraseña inválida (p.ej. 1234abcd)**
Salida esperada	El sistema web muestra un error indicando que no se cumple el patrón esperado
Salida obtenida	El usuario es informado de que su contraseña no cumple el patrón y posee más intentos para introducir de nuevo la credencial con la información suministrada
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.100: Caso de prueba - Indicar contraseña inválida (p.ej. 1234abcd)

Caso de prueba	UC-0101
Descripción	**Restablecer contraseña tras 30 minutos desde la recepción del *email***
Salida esperada	El sistema web detecta que el *token* ha expirado y el usuario es informado
Salida obtenida	El usuario intenta restablecer su contraseña pero consigue un error 404 (página no encontrada) debido a la expiración del *token*
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.101: Caso de prueba - Restablecer contraseña tras 30 minutos desde la recepción del email

Caso de prueba	UC-0102
Descripción	**Restablecer contraseña con enlace ya usado**
Salida esperada	El sistema web detecta que el *token* ha sido usado anteriormente y el usuario es informado
Salida obtenida	El usuario intenta restablecer su contraseña pero consigue un error 404 (página no encontrada) debido a que el *token* ya ha sido usado anteriormente
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.102: Caso de prueba - Restablecer contraseña con enlace ya usado

Caso de prueba	UC-0103
Descripción	**Dirección con protocolo HTTP (*Hypertext Transfer Protocol*)**
Salida esperada	El sistema web está limitado a comunicaciones seguras
Salida obtenida	La página no se muestra a través de una comunicación insegura
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.103: Caso de prueba - Dirección con protocolo HTTP (Hypertext Transfer Protocol)

Caso de prueba	UC-0104
Descripción	**Dirección con protocolo HTTPS (*Hypertext Transfer Protocol Secure*)**
Salida esperada	El sistema web carga la página sobre una comunicación segura
Salida obtenida	La página se muestra a través de una comunicación segura
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.104: Caso de prueba - Dirección con protocolo HTTPS (Hypertext Transfer Protocol Secure)

Caso de prueba	UC-0105
Descripción	**Introducir dirección inexistente**
Salida esperada	La dirección no existe y el usuario es notificado
Salida obtenida	El usuario consigue una página de error (404)
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.105: Caso de prueba - Introducir dirección inexistente

Caso de prueba	UC-0106
Descripción	**Introducir dirección sin poseer permiso de acceso**
Salida esperada	La página solicitada no se muestra al denegar el acceso
Salida obtenida	El usuario consigue una página de error (403)
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.106: Caso de prueba - Introducir dirección sin poseer permiso de acceso

Caso de prueba	UC-0107
Descripción	**Acceder al perfil personal del usuario**
Salida esperada	El sistema web muestra el perfil personal del usuario para su consulta y/o edición
Salida obtenida	El usuario se encuentra en su perfil personal con su información
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.107: Caso de prueba - Acceder al perfil personal del usuario

Caso de prueba	UC-0108
Descripción	**Modificar perfil personal del usuario**
Salida esperada	El sistema web actualiza el perfil y el usuario es notificado
Salida obtenida	El usuario actualiza su perfil correctamente (imagen, contraseña, etc.)
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.108: Caso de prueba - Modificar perfil personal del usuario

Caso de prueba	UC-0109
Descripción	**Error al modificar perfil**
Salida esperada	El sistema web no actualiza el perfil personal y el usuario es notificado
Salida obtenida	El usuario obtiene el mensaje asociado al error producido durante la modificación
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.109: Caso de prueba - Error al modificar perfil

Caso de prueba	UC-0110
Descripción	**Consultar estadísticas**
Salida esperada	El sistema web muestra las estadísticas actualizadas
Salida obtenida	El usuario administrador visualiza los gráficos estadísticos globales
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.110: Caso de prueba - Consultar estadísticas

Caso de prueba	UC-0111
Descripción	**Recargar gráfico estadístico**
Salida esperada	El sistema web recarga solamente el gráfico seleccionado
Salida obtenida	El usuario administrador visualiza el gráfico actualizado
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.111: Caso de prueba - Recargar gráfico estadístico

Caso de prueba	UC-0112
Descripción	**Listar usuarios administradores o usuarios normales**
Salida esperada	El listado de usuarios se muestra
Salida obtenida	El usuario administrador ve de forma ordenada y paginada todos los usuarios administradores o usuarios normales existentes en el sistema web
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.112: Caso de prueba - Listar usuarios administradores o usuarios normales

Caso de prueba	UC-0113
Descripción	**Listar usuarios administradores o usuarios normales visualizando determinados campos**
Salida esperada	Solamente los campos (columnas) seleccionados son visualizados en el listado de usuarios
Salida obtenida	El usuario administrador ve el listado de usuarios administradores o usuarios normales existentes en el sistema web pero solamente los campos seleccionados
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.113: Caso de prueba - Listar usuarios administradores o usuarios normales visualizando determinados campos

Caso de prueba	UC-0114
Descripción	**Buscar usuario administrador o usuario normal**
Salida esperada	El sistema web muestra únicamente el usuario buscado
Salida obtenida	El usuario administrador obtiene la información del usuario buscado
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.114: Caso de prueba - Buscar usuario administrador o usuario normal

Caso de prueba	UC-0115
Descripción	**Ordenar usuarios administradores o usuarios normales**
Salida esperada	El sistema web muestra el listado ordenado de forma ascendente o descendente
Salida obtenida	El usuario administrador obtiene la lista de usuarios administradores o usuarios normales ordenada según el campo elegido de forma ascendente o descendente
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.115: Caso de prueba - Ordenar usuarios administradores o usuarios normales

Caso de prueba	UC-0116
Descripción	**Exportar, imprimir o copiar usuarios administradores o usuarios normales**
Salida esperada	El sistema web exporta, imprime o copia todos los usuarios administradores o usuarios normales existentes
Salida obtenida	El usuario administrador obtiene un fichero PDF o CSV con todos los usuarios administradores o usuarios normales o presenta la posibilidad de impresión o copiado
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.116: Caso de prueba - Exportar, imprimir o copiar usuarios administradores o usuarios normales

Caso de prueba	UC-0117
Descripción	**Crear usuario administrador o usuario normal**
Salida esperada	El sistema web crea un nuevo usuario administrador o usuario normal con los datos introducidos
Salida obtenida	El nuevo usuario administrador o usuario normal ya tiene su cuenta disponible en el sistema web
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.117: Caso de prueba - Crear usuario administrador o usuario normal

Caso de prueba	UC-0118
Descripción	**Crear usuario administrador o usuario normal. Error**
Salida esperada	El sistema web no crea el nuevo usuario debido a un error y el usuario administrador es informado
Salida obtenida	El usuario administrador obtiene el error provocado durante la creación
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.118: Caso de prueba - Crear usuario administrador o usuario normal. Error

Caso de prueba	UC-0119
Descripción	**Comprobar disponibilidad de nombre de usuario o *email***
Salida esperada	El sistema web informa al usuario administrador si el nombre de usuario o *email* ha sido utilizado previamente y por tanto si está disponible o no
Salida obtenida	El usuario administrador conoce la disponibilidad del nombre de usuario o *email* introducido
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.119: Caso de prueba - Comprobar disponibilidad de nombre de usuario o email

Caso de prueba	UC-0120
Descripción	**Fortaleza de contraseña**
Salida esperada	El sistema web indica la fortaleza de la contraseña presentada
Salida obtenida	El usuario administrador obtiene información sobre si la contraseña introduce tiene una buena combinación de caracteres o no
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.120: Caso de prueba - Fortaleza de contraseña

Caso de prueba	UC-0121
Descripción	**Editar usuario administrador o usuario normal**
Salida esperada	El sistema web actualiza la información del usuario administrador o usuario normal
Salida obtenida	El usuario administrador edita un usuario administrador o usuario normal
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.121: Caso de prueba - Editar usuario administrador o usuario normal

Caso de prueba	UC-0122
Descripción	**Editar usuario administrador o usuario normal. Error**
Salida esperada	El sistema web no edita el usuario debido a un error
Salida obtenida	El usuario administrador obtiene el error provocado durante la edición
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.122: Caso de prueba - Editar usuario administrador o usuario normal. Error

Caso de prueba	UC-0123
Descripción	**Eliminar usuario administrador o usuario normal**
Salida esperada	El sistema web elimina el usuario correctamente
Salida obtenida	El usuario administrador elimina un usuario administrador o normal
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.123: Caso de prueba - Eliminar usuario administrador o usuario normal

Caso de prueba	UC-0124
Descripción	**Eliminar usuario administrador o usuario normal. Error**
Salida esperada	El sistema web no elimina el usuario debido a un error y el usuario administrador es informado
Salida obtenida	El usuario administrador obtiene el error de la eliminación
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.124: Caso de prueba - Eliminar usuario administrador o usuario normal. Error

Caso de prueba	UC-0125
Descripción	**Visualizar información de monitorización con usuario administrador**
Salida esperada	El sistema web muestra la información completa de monitorización
Salida obtenida	El usuario administrador obtiene las mediciones y alertas de todos los participantes
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.125: Caso de prueba - Visualizar información de monitorización con usuario administrador

Caso de prueba	UC-0126
Descripción	**Visualizar información de monitorización con usuario normal**
Salida esperada	El sistema web muestra la información de monitorización de la que el usuario normal es poseedor
Salida obtenida	El usuario normal obtiene las mediciones y alertas tomadas con ese mismo nombre de usuario como participante. Además, visualiza sus estadísticas
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.126: Caso de prueba - Visualizar información de monitorización con usuario normal

Caso de prueba	UC-127
Descripción	**Sistema web en idioma inglés**
Salida esperada	El idioma de la aplicación cambia a inglés
Salida obtenida	El usuario interactúa con la aplicación en inglés
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.127: Caso de prueba - Sistema web en idioma inglés

Caso de prueba	UC-0128
Descripción	**Probar el sistema web en diferentes navegadores**
Salida esperada	El sistema web permite la navegación correcta en navegadores con diferentes motores de renderizado
Salida obtenida	El usuario visualiza correctamente el sistema en diferentes navegadores
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.128: Caso de prueba - Probar el sistema web en diferentes navegadores

Caso de prueba	UC-0129
Descripción	**Probar el sistema web en diferentes dispositivos electrónicos**
Salida esperada	El sistema web adapta la interfaz al tamaño de pantalla
Salida obtenida	El usuario visualiza correctamente el sistema en el dispositivo
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.129: Caso de prueba - Probar el sistema web en diferentes dispositivos electrónicos

Caso de prueba	UC-0130
Descripción	**Visualizar *email* ***
Salida esperada	*Email* enviado por el sistema web es recibido y visualizado correctamente
Salida obtenida	El formato y estilo del *email* se visualiza correctamente en el gestor de correos
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.130: Caso de prueba - Visualizar email

6.2.5. Componente - Protocolo de registro de incidencias

Los casos de prueba a continuación descritos hacen referencia al componente del proyecto Hyot que define el modelo de la red de negocio desplegada en la BC de HF. Para la ejecución de estos casos de prueba se utilizó la herramienta *playground* de HC [33].

Caso de prueba	UC-0131
Descripción	**Registrar transacción *PublishAlert* con participante de tipo *User***
Salida esperada	La transacción se registra en la BC
Salida obtenida	El participante registra una transacción creando un activo *Alert*
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.131: Caso de prueba - Registrar transacción PublishAlert con participante de tipo User

Caso de prueba	UC-0132
Descripción	**Crear activo *Alert* con participante de tipo *User***
Salida esperada	La operación se deniega
Salida obtenida	El participante no puede registrar directamente un activo *Alert*
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.132: Caso de prueba - Crear activo Alert con participante de tipo User

Caso de prueba	UC-0133
Descripción	**Obtener activos *Alert* de los que el participante de tipo *User* es poseedor**
Salida esperada	La consulta se ejecuta correctamente retornando los activos *Alert*
Salida obtenida	El participante obtiene todos los activos *Alert* registrados por él mismo en la BC
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.133: Caso de prueba - Obtener activos Alert de los que el participante de tipo User es poseedor

Caso de prueba	UC-0134
Descripción	**Obtener activos *Alert* de los que el participante de tipo *User* no es poseedor**
Salida esperada	La operación se deniega
Salida obtenida	El participante solamente puede obtener los activos *Alert* cuyo poseedor sea él mismo
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.134: Caso de prueba - Obtener activos Alert de los que el participante de tipo User no es poseedor

Caso de prueba	UC-0135
Descripción	**Editar activo *Alert* con participante de tipo *User***
Salida esperada	La operación se deniega
Salida obtenida	El participante no puede editar un activo *Alert* ya sea de su posesión o de posesión de otro participante de tipo *User*
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.135: Caso de prueba - Editar activo Alert con participante de tipo User

Caso de prueba	UC-0136
Descripción	**Eliminar activo *Alert* con participante de tipo *User***
Salida esperada	La operación se deniega
Salida obtenida	El participante no puede eliminar un activo *Alert* ya sea de su posesión o de posesión de otro participante de tipo *User*
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.136: Caso de prueba - Eliminar activo Alert con participante de tipo User

Caso de prueba	UC-0137
Descripción	**Crear, editar o eliminar participante de tipo *User* con un participante de tipo *User***
Salida esperada	La operación se deniega
Salida obtenida	El participante no presenta el privilegio de realizar alguna acción sobre participantes de tipo *User*
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.137: Caso de prueba - Crear, editar o eliminar participante de tipo User con un participante de tipo User

Caso de prueba	UC-0138
Descripción	**Gestionar identidades con participante de tipo *User***
Salida esperada	La operación se deniega
Salida obtenida	El participante no puede gestionar ni enlazar identidades
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.138: Caso de prueba - Gestionar identidades con participante de tipo User

Caso de prueba	UC-0139
Descripción	**Acciones con participante administrador**
Salida esperada	Cualquier operación -crear, editar y eliminar activos o participantes, registrar transacciones, gestionar y enlazar identidades, desplegar la red de negocio, etc.- realizada por este tipo de participante se procesa correctamente
Salida obtenida	El participante de tipo administrador presenta acceso completo a los recursos de sistema y de usuario
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.139: Caso de prueba - Acciones con participante administrador

Caso de prueba	UC-0140
Descripción	**Enviar consultas con participante administrador para obtener información filtrada**
Salida esperada	La consulta se ejecuta retornando la información filtrada
Salida obtenida	El participante de tipo administrador obtiene información filtrada (p.ej. alertas originadas por cada sensor y cada evento, alertas originadas por cada usuario, alertas originadas por un determinado sensor y/o evento y usuario, etc.) a partir de enviar una consulta parametrizada
Estado	Finalizado
Resultado	**Correcto**

Tabla 6.140: Caso de prueba - Enviar consultas con participante administrador para obtener información filtrada

Conclusiones y trabajo futuro

Conclusiones

El camino hacia la Industria 4.0 o *Industria inteligente* llevada de la mano por el Internet de las Cosas (*Internet of Things* -IoT-) entre otras áreas, pasa por digitalizar y sustentar las necesidades reales de los clientes con las nuevas tecnologías. Esta rápida evolución del mercado del IoT ha provocado una explosión en cuanto al número y variedad de soluciones IoT se refiere, lo que ha creado grandes desafíos a medida que la industria evoluciona. Principalmente, la necesidad urgente de un modelo seguro y con un grado creciente de confianza y protección de la privacidad para realizar tareas tan comunes como detección, almacenamiento y comunicación de información. La prueba de la existencia, de la originación o de un registro consecuente de la pista en un marco temporal gana la importancia y es por ello que la aplicación de la tecnología Blockchain (BC) puede favorecer el despliegue de soluciones y arquitecturas más seguras, confiables y sin intermediación de terceros frente a otros mecanismos tradicionales de persistencia.

Con este objetivo surge Hyot, un proyecto *software* de código abierto[3] que propone una prueba de concepto (*Proof of Concept* -PoC-) simple y novedosa sobre conceptos y tecnologías en auge como la BC. Con ello, se pretende mostrar las ventajas de las arquitecturas basadas en esta tecnología a la hora de diseñar e implementar protocolos de control y auditoría en IoT y cómo mediante este mecanismo se puede preservar la autenticidad de ocurrencia de un hecho en un instante temporal dado.

Por último, mencionar que se trata de un proyecto englobado dentro de un área con gran interés puesto que fue aceptado en la categoría de *Investigación en desarrollo* de las **IV Jornadas Nacionales de Investigación en Ciberseguridad**[4], celebradas en los días 13-15 de junio de 2018, en Donostia-San Sebastián. Además, el ciclo de vida de este proyecto no pretende ser el habitual de un trabajo final de universidad ya que la idea principal es continuar con su desarrollo una vez presentado -siguiendo las líneas de trabajo futuro propuestas a continuación- y poder presentarlo a otros congresos y/o publicarlo en alguna revista científica.

[3]El código fuente se encuentra liberado en la plataforma Github. Consulte el anexo G para obtener más información.

[4]Puede consultar el artículo enviado en las actas del congreso en [48].

Trabajo futuro

El proyecto desarrollado, Hyot, aunque es 100 % funcional y válido para una primera versión beta no se puede considerar una PoC totalmente completa y finalizada sino más bien un desarrollo inicial y experimental con tecnologías en auge hoy en día utilizadas en infinidad de sectores. Debido al carácter de este trabajo y por tanto la existencia de la limitación temporal para su desarrollo se implementó la funcionalidad núcleo, identificando a mayores una serie de líneas marcadas como trabajo futuro con el fin de perfeccionar la solución y dotarla de mayor funcionalidad. A continuación, las líneas de trabajo futuro identificadas:

- Funcionalidades de gestión de la identidad y su equilibrio con la protección de la privacidad [42], [43] y con la interoperabilidad con BC públicas [37], [86].
- Incorporación de técnicas de aprendizaje automático que permitan la detección de objetos y personas con el fin de identificar información sensible y relevante como evidencia digital [73].
- Cumplimiento de un alto estándar de seguridad por medio de almacenar las credenciales GPG (*GNU Privacy Guard*) en una partición o un dispositivo externo (p.ej. USB o disco duro) cifrado.
- Ampliar la funcionalidad de los componentes y sistema web como, por ejemplo:
  - Registro de la existencia de incidencias en un sistema descentralizado de sellado de tiempo que utiliza la Blockchain pública Bitcoin [67][5].
  - Intercomunicación entre el componente de monitorización de sucesos del entorno y el sistema web a través de la plataforma IoT de IBM Cloud para la emisión y suscripción de comandos tal como, encendido y apagado de la monitorización.
  - Notificación de alertas por otros medios, tal como SMS, Telegram, etc.
  - Uso del código QR para la desencriptación.
  - Autenticación multiusuario en el servidor expuesto por Hyperledger Composer (HC) para que cada transacción se encuentre firmada por el propio poseedor.
  - Ampliar la funcionalidad incluyendo nuevos componentes electrónicos en el prototipo: sensor de movimiento, zumbador (*buzzer*), etc.

[5]Esta funcionalidad requiere de Python 3 por lo que se valorará la adaptación del código a dicha versión.

Bibliografía

[1] **All About Circuits;** "RESISTOR COLOR CODES"
https:
//www.allaboutcircuits.com/textbook/reference/chpt-2/resistor-color-codes/
Fecha última consulta: 11 de agosto de 2018.

[2] **Ambrosus;**
https://ambrosus.com/
Fecha última consulta: 12 de diciembre de 2017.

[3] **Androulaki E., Cachin C., Ferris C., Muralidharan S., Murthy C., Nguyen B., Sethi M. & Stathakopoulou C.;** "HYPERLEDGER FABRIC: A DISTRIBUTED OPERATING SYSTEM FOR PERMISSIONED BLOCKCHAINS". April, 2018. arXiv:1801.10228v2.
Fecha última consulta: 14 de agosto de 2018.

[4] **Atzei N., Bartoletti M. & Cimoli T.;** "A SURVEY OF ATTACKS ON ETHEREUM SMART CONTRACTS (SOK)". March, 2017.
Fecha última consulta: 20 de febrero de 2018.

[5] **Azouvi S., Maller M. & Meiklejohn S.;** "EGALITARIAN SOCIETY OR BENEVOLENT DICTATORSHIP: THE STATE OF CRYPTOCURRENCY GOVERNANCE". The Fifth Workshop on Bitcoin and Blockchain Research. 2018.
Fecha última consulta: 14 de mayo de 2018.

[6] **BigchainDB;** https://www.bigchaindb.com/
Fecha última consulta: 12 de enero de 2018.

[7] **Bitfury;** http://bitfury.com/
Fecha última consulta: 07 de diciembre de 2017.

[8] **Buschmann F., Meunier R., Rohnert H., Sommerlad P. & Stal M.;** "PATTERN-ORIENTED SOFTWARE ARCHITECTURE - A SYSTEM OF PATTERNS". Wiley and Sons. Pages 125-144. July, 1996.
Fecha última consulta: 20 de febrero de 2018.

[9] **CA/Browser Forum;** "MEMBERS"
https://cabforum.org/members/
Fecha última consulta: 17 de julio de 2018.

[10] **Christidis K. & Devetsikiotis M.;** "BLOCKCHAINS AND SMART CONTRACTS FOR THE INTERNET OF THINGS". IEEE Access. June, 2016. DOI: 10.1109/ACCESS.2016.2566339.
Fecha última consulta: 03 de agosto de 2018.

[11] **ChromaWay;** https://chromaway.com/
Fecha última consulta: 06 de diciembre de 2017.

[12] **Conoscenti M., Vetrò A. & De Martin, J.C.;** "BLOCKCHAIN FOR THE INTERNET OF THINGS: A SYSTEMATIC LITERATURE REVIEW". IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA). November, 2016. DOI: 10.1109/AICCSA.2016.7945805.
Fecha última consulta: 25 de abril de 2018.

[13] **Croxton M.;** "GENERATE SSL CERTIFICATES WITH SUBJECT ALT NAMES ON OS X"
https://gist.github.com/croxton/ebfb5f3ac143cd86542788f972434c96
Fecha última consulta: 03 de julio de 2018.

[14] **DARPA;**
https://www.darpa.mil/
Fecha última consulta: 18 de febrero de 2018.

[15] **Dorri A., Kanhere S. & Jurdak R.;** "TOWARDS AN OPTIMIZED BLOCKCHAIN FOR IOT". In Proceedings of the Second International Conference on Internet-of-Things Design and Implementation. ACM. Pages 173-178. April, 2017. DOI: 10.1145/3054977.3055003.
Fecha última consulta: 28 de abril de 2018.

[16] **Douceur J. R.;** "THE SYBIL ATTACK". In Peer-to-Peer Systems. Lecture Notes in Computer Science book series. Springer. Volume-2429. Pages 251-260. October, 2002.
Fecha última consulta: 16 de diciembre de 2017.

[17] **Dwork C. & Naor M.;** "PRICING VIA PROCESSING, OR, COMBATTING JUNK MAIL, ADVANCES IN CRYPTOLOGY". CRYPTO'92. Lecture Notes in Computer Science. No. 740. Springer: 139–147. 1993.
Fecha última consulta: 27 de enero de 2018.

[18] **Ethereum;**
https://www.ethereum.org/
Fecha última consulta: 26 de junio de 2018.

[19] **Everledger;**
https://www.everledger.io/
Fecha última consulta: 07 de diciembre de 2017.

[20] **Evernym;** https://www.evernym.com/
Fecha última consulta: 20 de enero de 2018.

[21] **Factom;** https://www.factom.com/
Fecha última consulta: 07 de diciembre de 2017.

[22] **Franco P.;** "UNDERSTANDING BITCOIN: CRYPTOGRAPHY, ENGINEERING AND ECONOMICS". Wiley Finance Series. November, 2014.
Fecha última consulta: 28 de agosto de 2018.

[23] **Gamma E., Helm R., Johnson R., Vlissides J. & Booch G.;** "DESIGN PATTERNS: ELEMENTS OF REUSABLE OBJECT-ORIENTED SOFTWARE". Addison-Wesley Professional. First edition. Pages 127-134. November, 1994.
Fecha última consulta: 21 de febrero de 2018.

[24] **GnuPG;**
https://www.gnupg.org/download/index.html
Fecha última consulta: 16 de mayo de 2018.

[25] **GNU Bash;**
https://www.gnu.org/software/bash/
Fecha última consulta: 03 de febrero de 2018.

[26] **Grails;**
https://grails.org/
Fecha última consulta: 29 de agosto de 2018.

[27] **Guardtime;**
https://guardtime.com/
Fecha última consulta: 16 de enero de 2018.

[28] **Hashcash;**
http://www.hashcash.org/
Fecha última consulta: 22 de julio de 2018.

[29] **Hyperledger;**
https://www.hyperledger.org/
Fecha última consulta: 12 de abril de 2018.

[30] **Hyperledger Architecture. Volume 1;**
https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger_Arch_WG_Paper_1_Consensus.pdf
Fecha última consulta: 12 de agosto de 2018.

[31] **Hyperledger Composer;**
https://www.hyperledger.org/projects/composer
Fecha última consulta: 12 de abril de 2018.

[32] **Hyperledger Composer - Access Control Language;**
https://hyperledger.github.io/composer/latest/reference/acl_language
Fecha última consulta: 22 de mayo de 2018.

[33] **Hyperledger Composer - Playground;**
https://composer-playground.mybluemix.net/login
Fecha última consulta: 17 de julio de 2018.

[34] **Hyperledger Composer - Releases;**
https://github.com/hyperledger/composer/releases
Fecha última consulta: 28 de mayo de 2018.

[35] **Hyperledger Fabric;**
https://www.hyperledger.org/projects/fabric
Fecha última consulta: 22 de agosto de 2018.

[36] **Hyperledger Fabric Roadmap;**
https://wiki.hyperledger.org/projects/fabric/roadmap
Fecha última consulta: 05 de junio de 2018.

[37] **Hyperledger Sawtooth;**
https://www.hyperledger.org/projects/sawtooth
Fecha última consulta: 26 de agosto de 2018.

[38] **Iansiti M. & Lakhani R. K.;** "THE TRUTH ABOUT BLOCKCHAIN". Harvard Business Review. Published in HBR. January-February, 2017.
Fecha última consulta: 27 de julio de 2018.

[39] **IBM Blockchain;** "DEVELOP IN A CLOUD SANDBOX IBM BLOCKCHAIN PLATFORM".
https://ibm-blockchain.github.io/
Fecha última consulta: 13 de agosto de 2018.

[40] **IBM Cloud;**
https://www.ibm.com/cloud-computing/es/es/
Fecha última consulta: 02 de septiembre de 2018.

[41] **IBM Cloud - Blockchain;**
https://console.bluemix.net/catalog/services/blockchain
Fecha última consulta: 13 de agosto de 2018.

[42] **IBM Research - Zurich;** "IDENTITY MIXER"
https://www.zurich.ibm.com/identity_mixer/
Fecha última consulta: 29 de agosto de 2018.

[43] **Idemix (Identity Mixer) en Hyperledger Fabric;**
https://jira.hyperledger.org/browse/FAB-2005
Fecha última consulta: 29 de agosto de 2018.

[44] **Incibe;** "GESTIÓN DE RIESGOS - UNA GUÍA DE APROXIMACIÓN PARA EL EMPRESARIO"
https://www.incibe.es/sites/default/files/contenidos/guias/doc/guia_ciberseguridad_gestion_riesgos_metad.pdf
Fecha última consulta: 04 de enero de 2018.

[45] **Incibe;** "GUÍA AVANZADA DE GESTIÓN DE RIESGOS"
Fecha última consulta: 29 de diciembre de 2017.

[46] **ISO;** "ISO 27001:2013 INFORMATION SECURITY MANAGEMENT SYSTEMS - REQUIREMENTS"
http://www.iso.org/iso/home/standards/management-standards/iso27001.htm
Fecha última consulta: 29 de diciembre de 2017.

[47] **JavaScript;**
https://www.javascript.com/
Fecha última consulta: 20 de mayo de 2018.

[48] **JNIC 2018;** "ACTAS DE LAS IV JORNADAS NACIONALES DE INVESTIGACIÓN EN CIBERSEGURIDAD". Mondragon Unibertsitatea. Páginas 111-112. Junio, 2018.
http://2018.jnic.es/assets/Actas_JNIC2018.pdf
Fecha última consulta: 15 de julio de 2018.

[49] **Journal Transcript;** "BITCOIN MINING CONSUMES AS MUCH ELECTRICITY AS THE NATION OF IRELAND"
https://www.journaltranscript.com/2018/06/bitcoin-mining-consumes-as-much-electricity-as-the-nation-of-ireland/
Fecha última consulta: 27 de junio de 2018.

[50] **Karame G., Androulaki E. & Capkun S.;** "TWO BITCOINS AT THE PRICE OF ONE? DOUBLE-SPENDING ATTACKS ON FAST PAYMENTS IN BITCOIN". Proceedings of the 2012 ACM conference on Computer and communications security. October, 2012. DOI:

10.1145/2382196.2382292.
Fecha última consulta: 03 de agosto de 2018.

[51] **Konstantinos C. & Devetsikiotis M.;** "BLOCKCHAINS AND SMART CONTRACTS FOR THE INTERNET OF THINGS". IEEE. May, 2016. DOI: 10.1109/ACCESS.2016.2566339.
Fecha última consulta: 20 de marzo de 2018.

[52] **Kontakt;**
https://kontakt.io/
Fecha última consulta: 11 de diciembre de 2017.

[53] **Krishnan V.;** "CREATING AN UML DIAGRAM FROM HYPERLEDGER COMPOSER MODEL"
https://medium.com/@vkrishnan.ny/
creating-an-uml-diagram-from-hyperledger-composer-model-e2ce42ad257
Fecha última consulta: 06 de julio de 2018.

[54] **Laszka A., Dubey A., Walker M. & Schmidt D.;** "PROVIDING PRIVACY, SAFETY, AND SECURITY IN IOT-BASED TRANSACTIVE ENERGY SYSTEMS USING DISTRIBUTED LEDGERS". In Proceedings of the Seventh International Conference on the Internet of Things. ACM. September, 2017. DOI: 10.1145/3131542.3131562.
Fecha última consulta: 11 de diciembre de 2017.

[55] **Levy K.;** "BOOK-SMART, NOT STREET-SMART: BLOCKCHAIN-BASED SMART CONTRACTS AND THE SOCIAL WORKINGS OF LAW". Engaging Science, Technology, and Society 3. 2017. DOI: 10.17351/ests2017.107.
Fecha última consulta: 14 de junio de 2018.

[56] **Litecoin;**
https://litecoin.org/
Fecha última consulta: 20 de enero de 2018.

[57] **Madhuravani B. & Murthy R. S. D.;** "CRYPTOGRAPHIC HASH FUNCTIONS: SHA FAMILY". International Journal of Innovative Technology and Exploring Engineering (IJITEE). Volume-2, Issue-4. Pages 326-329. March, 2013. ISSN: 2278-3075.
Fecha última consulta: 22 de marzo de 2018.

[58] **MaidSafe;**
https://maidsafe.net/
Fecha última consulta: 12 de diciembre de 2017.

[59] **Menezes A., Van Oorschot P. C. & Vanstone S.;** "HANDBOOK OF APPLIED CRYPTOGRAPHY". CRC Press. October, 1996.
Fecha última consulta: 11 de julio de 2018.

[60] **Microsoft Azure - Blockchain;**
https://azure.microsoft.com/es-es/solutions/blockchain/
Fecha última consulta: 13 de agosto de 2018

[61] **Mimura Gonzalez N., Miers C., Frota Redígolo F., Simplício M., Carvalho T., Näslund M. & Pourzandi M.;** "A TAXONOMY MODEL FOR CLOUD COMPUTING SERVICES". In Proceedings of the 1st International Conference on Cloud Computing and Services Science. Pages 56-65. January, 2011. DOI: 10.5220/0003384800560065.
Fecha última consulta: 18 de marzo de 2018.

[62] **Multichain - Permissions management;**
https://www.multichain.com/developers/permissions-management/
Fecha última consulta: 15 de diciembre de 2017.

[63] **Nakamoto S.;** "BITCOIN: A PEER-TO-PEER ELECTRONIC CASH SYSTEM"
https://bitcoin.org/bitcoin.pdf
Fecha última consulta: 17 de julio de 2018.

[64] **Narayanan A. & Clark J.;** "THE CONCEPT OF CRYPTOCURRENCIES IS BUILT FROM FORGOTTEN IDEAS IN RESEARCH LITERATURE."
https://queue.acm.org/detail.cfm?id=3136559
Fecha última consulta: 06 de marzo de 2018.

[65] **Nokia Networks;** "SENSING AS A SERVICE"
https://networks.nokia.com/services/sensing-as-a-service
Fecha última consulta: 06 de diciembre de 2017.

[66] **Novo O.;** "BLOCKCHAIN MEETS IOT: AN ARCHITECTURE FOR SCALABLE ACCESS MANAGEMENT IN IOT". Journal of Internet of Things class files. Vol. 14, N.8. March, 2018.
Fecha última consulta: 10 de julio de 2018.

[67] **OpenTimestamps;**
https://opentimestamps.org/
Fecha última consulta: 17 de agosto de 2018.

[68] **Pérez-Solà C. & Herrera-Joancomartí J.;** "BITCOINS Y EL PROBLEMA DE LOS GENERALES BIZANTINOS". RECSI. Septiembre, 2014.
Fecha última consulta: 16 de mayo de 2018.

[69] **Python;**
https://www.python.org/
Fecha última consulta: 12 de abril de 2018.

[70] **Python;** "requests http for humans - advanced usage"
http:
//docs.python-requests.org/en/latest/user/advanced/#ssl-cert-verification
Fecha última consulta: 16 de abril de 2018.

[71] **Qualys SSL Labs;** "ssl and tls deployment best practices"
https:
//github.com/ssllabs/research/wiki/SSL-and-TLS-Deployment-Best-Practices
Fecha última consulta: 18 de agosto de 2018.

[72] **R3;**
https://www.r3.com/
Fecha última consulta: 14 de diciembre de 2017.

[73] **Ramachandran A. & Kantarcioglu M.;** "smartprovenance: a distributed, blockchain based data provenance system". In Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy. ACM. Pages 35-42. March, 2018. DOI: 10.1145/3176258.3176333.
Fecha última consulta: 22 de agosto de 2018.

[74] **Raspberry Pi Pinout;**
https://es.pinout.xyz/
Fecha última consulta: 03 de abril de 2018.

[75] **Ripple;**
https://ripple.com/
Fecha última consulta: 14 de diciembre de 2017.

[76] **Risk Management;** "metrics and measurement problems quantitative risk management"
http://management-of-risk.blogspot.com.es/
Fecha última consulta: 29 de diciembre de 2017.

[77] **Samaniego M. & Deters R.;** "internet of smart things – iost". IEEE 1st International Conference on Cognitive Computing. 2017. DOI: 10.1109/IEEE.ICCC.2017.9.
Fecha última consulta: 11 de diciembre de 2017.

[78] **Stackoverflow;** "getting chrome to accept self-signed localhost certificate"
https://stackoverflow.com/questions/7580508/
getting-chrome-to-accept-self-signed-localhost-certificate
Fecha última consulta: 03 de julio de 2018.

[79] **Stampery;**
https://stampery.com/
Fecha última consulta: 05 de diciembre de 2017.

[80] **Szabo N.;** "formalizing and securing relationships on public networks". First Monday. Vol.2, N.9. September, 1997.
Fecha última consulta: 23 de diciembre de 2017.

[81] **Torrecilla P.;** "proceso unificado ágil: disciplinas y fases"
http://nosolopau.com/2012/06/07/
mas-sobre-el-proceso-unificado-agil-fases-y-disciplinas/
Fecha última consulta: 26 de diciembre de 2017.

[82] **Ultrasonic Ranging Module HC-SR04;** "datasheet"
https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf
Fecha última consulta: 17 de julio de 2018.

[83] **UML;** "the unified modeling language"
https://www.uml-diagrams.org/
Fecha última consulta: 25 de febrero de 2018.

[84] **Velox.re;** https://www.velox.re/
Fecha última consulta: 09 de diciembre de 2017.

[85] **Wattenhofer R.;** "distributed ledger technology: the science of the Blockchain". Inverted Forest Publishing. Second Revised Edition. March, 2017.
Fecha última consulta: 28 de junio de 2018.

[86] **Xu R., Chen Y., Blasch E. & Chen G.;** "blendcac: a blockchain-enabled decentralized capability-based access control for iots". April, 2018. ArXiv:1804.09267v1.
Fecha última consulta: 28 de agosto de 2018.

[87] **Xu X., Weber I., Staples M., Zhu L., Bosch J., Bass L., Pautasso C. & Rimba P.;** "a taxonomy of blockchain-based systems for architecture design". IEEE International Conference on Software Architecture. April, 2017. DOI: 10.1109/ICSA.2017.33 .
Fecha última consulta: 16 de diciembre de 2017.

[88] **Zahnentferner J.;** "chimeric ledgers: translating and unifying utxo-based and account-based cryptocurrencies". 2018.
Fecha última consulta: 16 de diciembre de 2017.

[89] **Zensar;** "REINVENTING MORTGAGE WITH BLOCKCHAIN".
http://www.zensar.com/blogs/2017/05/reinventing-mortgage-with-blockchain/
Fecha última consulta: 11 de diciembre de 2017.

[90] **Zheng Z., Xie S., Dai H., Chen X. & Wang H.;** "BLOCKCHAIN CHALLENGES AND OPPORTUNITIES: A SURVEY". Inderscience Enterprises Ltd. 2018.
Fecha última consulta: 20 de marzo de 2018.

[91] **Wüst K. & Gervais A.;** "DO YOU NEED A BLOCKCHAIN?". 2017.
Fecha última consulta: 05 de marzo de 2018.

ANEXOS

Anexo A

Material empleado

El contexto de desarrollo de un proyecto está compuesto por todas aquellas herramientas que han estado involucradas en el desarrollo de éste, englobando tanto herramientas de tipo *software* como de tipo *hardware*. Para el presente proyecto, éstas han sido las siguientes:

- Herramientas de tipo *software*:
  - DROPBOX: herramienta para el almacenamiento en la nube de activos resultantes de la ejecución del proyecto.
  - FRITZING: herramienta para el diseño de prototipos electrónicos.
  - GITHUB: servicio para alojamiento de repositorios *software* gestionado por el sistema de control de versiones Git.
  - GITKRAKEN: cliente Git utilizado.
  - IBM CLOUD: conjunto de servicios de computación en la nube ofrecidos por la compañía tecnológica IBM.
  - INTELLIJ IDEA: IDE (*Integrated Development Environment*)„ desarrollado por la empresa JetBrains, para el entorno Java aunque soporta también otros lenguajes. Se utilizó para desarrollar el sistema web.
  - LUCIDCHART: herramienta en línea para la elaboración de diagramas.
  - MÁQUINA VIRTUAL (*Virtual Machine* -VM-): desplegada en la plataforma Microsoft Azure con el sistema operativo Ubuntu. Se empleará para desplegar Hyperledger Fabric (HF) y Hyperledger Composer (HC).
  - MICROSOFT AZURE: plataforma informática en la nube de nivel empresarial abierta y flexible ofrecida por la compañía tecnológica Microsoft.
  - MICROSOFT POWERPOINT 2016: herramienta utilizada para la elaboración de la presentación de este Trabajo Fin de Máster (TFM).

- MICROSOFT PROJECT 2016: herramienta utilizada para la planificación del proyecto, con el fin de llevar a cabo un seguimiento detallado de la evolución de éste y de la carga de trabajo.
- NGROK: herramienta de tunelización segura hacia servidores locales.
- NODE PACKAGE MANAGER (NPM): manejador de paquetes por defecto para Node.js.
- NODE VERSION MANAGER (NVM): gestor de versiones para Node.js.
- ONEDRIVE: herramienta para el almacenamiento en la nube de los documentos relativos a este TFM.
- PYCHARM: IDE, desarrollado por la empresa JetBrains, para el entorno Python. Se utilizó para desarrollar los componentes de Python.
- REM: herramienta para documentar la gestión de requisitos.
- SEQUEL PRO: herramienta para gestionar bases de datos (BBDD) MySQL.
- TEXPAD: herramienta para documentar la memoria del TFM en LaTeX.
- TRELLO: herramienta para la gestión y organización de proyectos.
- VISUAL STUDIO CODE: editor de código, desarrollado por la empresa Microsoft, empleado para la creación del componente en lenguaje *Bash* (*Bourne-again Shell*) y para la definición de la red de negocio con la herramienta de desarrollo HC. Para el segundo caso, se utilizó este editor debido a la existencia de un *plugin* instalable.

- Herramientas de tipo *hardware*:

  - RASPBERRY PI: Computador de placa única (*Single Board Computer* -SBC-) empleado como dispositivo principal para la implantación del proyecto.
  - CÁMARA PARA RPI: dispositivo utilizado para la captura de vídeos.
  - SENSORES: elementos para capturar información del entorno exterior.
  - DISPOSITIVOS DE SALIDA: elementos para informar del estado durante la ejecución del programa informático, donde se ubican: el LCD (*Liquid Crystal Display*) y el LED (*Light-Emitting Diode*).
  - PLACA DE PRUEBAS (BREADBOARD), PLACA DE EXTENSIÓN GPIO (*General Purpose Input Output*) DE TIPO T Y CABLEADO: para conectar los diferentes elementos que conforman el circuito electrónico.
  - SAMSUNG GALAXY TAB S 10,5: dispositivo utilizado para la comprobación visual y funcional del sistema web en dispositivos con sistema operativo (SO): Android.
  - IPHONE 6 PLUS: dispositivo utilizado para la comprobación visual y funcional del sistema web en dispositivos con SO: iOS.

- MACBOOK PRO MID 2014: ordenador principal para el desarrollo del proyecto. Posee MacOS (*Macintosh Operating System*) High Sierra como SO y cuenta con las siguientes características: Intel Core i7 a 2,2GH con 16 GB 1600 MHz DDR3 de memoria RAM (*Random Access Memory*) y disco duro SSD (*Solid-State Drive*).

Anexo B

Planificación del proyecto

Un adecuado proceso de desarrollo *software* y gestión del proyecto son esenciales en la actualidad para incrementar la probabilidad de éxito y la calidad final de un proyecto. En este capítulo se define la planificación y organización del presente proyecto junto con la estimación de costes y la evaluación de riesgos.

B.1. Proceso de desarrollo software

Las metodologías y estándares utilizados en el desarrollo *software* proporcionan las pautas básicas para poder conocer el camino completo a recorrer desde antes de empezar la implementación, favoreciendo la probabilidad de obtener calidad en el producto final, así como el cumplimiento en la entrega del mismo en el tiempo estipulado.

Es de vital importancia elegir la metodología adecuada, así como las herramientas de implementación adecuadas. Para el desarrollo del presenta proyecto se han seguido las directrices de la metodología de desarrollo *software*: **Proceso Unificado Ágil (*Agile Unified Process* -AUP-)**, definido por Scott Ambler como un marco de trabajo genérico que representa una versión simplificada del proceso unificado racional (*Rational Unified Process* -RUP-) que combina conceptos propios de éste con técnicas ágiles con el objetivo de proporcionar las bases para mejorar la productividad.

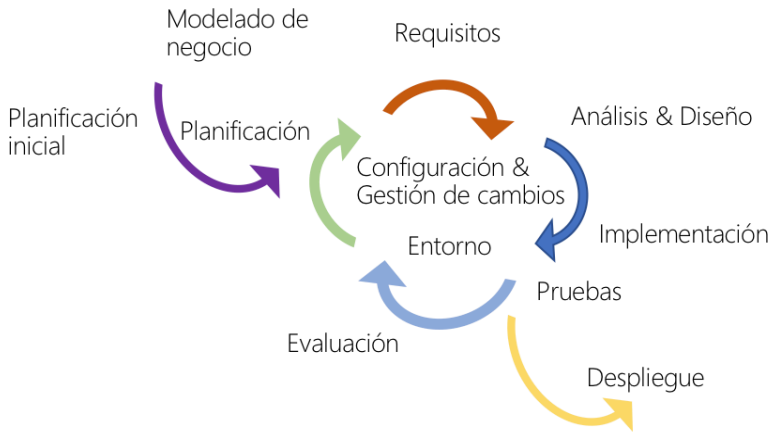
El motivo de la elección de esta metodología está estrechamente relacionada con el enfoque intermedio entre dos tipos de metodologías diferentes:

- eXtreme Programming (XP): metodología ágil de desarrollo *software* que se centra en mayor medida en la adaptabilidad más que en la previsibilidad. Su gran inconveniente es que para determinados desarrolladores este proceso es demasiado ligero, es decir, no especifica cómo crear algunos artefactos necesarios.

- Proceso Unificado Racional (RUP): proceso de desarrollo *software* iterativo e incremental centrado en la arquitectura, dirigido por los casos de uso y enfocado en los riesgos. Al ser iterativo e incremental ayuda a mitigar los posibles riesgos así como descubrir nuevos errores y pulirlos. Su gestión resulta realmente sencilla, sin embargo la gran desventaja que presenta es la cantidad de artefactos que requiere a los desarrolladores.

El ciclo de vida AUP (Figura B.1) es una implementación de desarrollo en espiral. Se compone de cuatro fases cada cual concluye con una versión intermedia. Al terminar cada fase se realiza una evaluación para determinar si se ha cumplido o no con los objetivos de la misma.

Figura B.1: Ciclo de vida de AUP.



Las cuatro fases en la que la metodología AUP divide su ciclo de desarrollo, son:

- Fase de inicio: esta fase tiene como propósito definir y acordar la visión y alcance del proyecto con el cliente, identificar los posibles riesgos asociados al proyecto y proponer una planificación temporal.

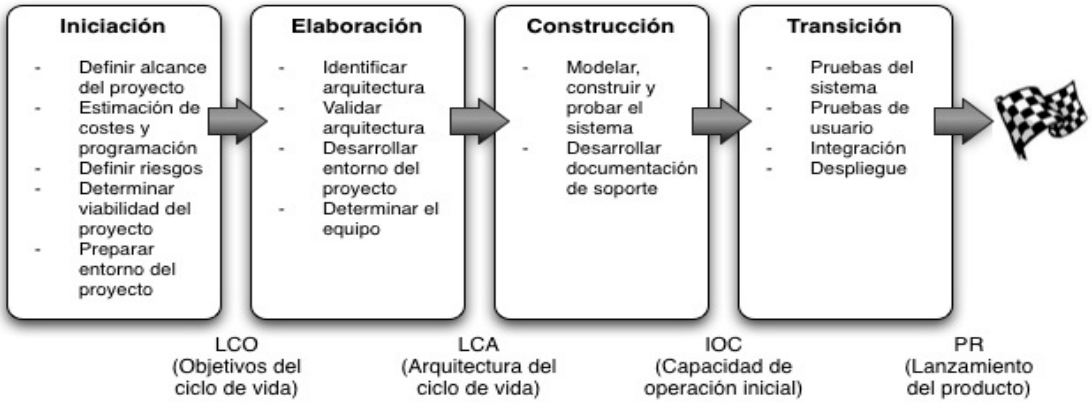
- Fase de elaboración: se identifican, y posteriormente, se detallan los requisitos y casos de uso que permiten definir la arquitectura base del sistema, el modelo de dominio y las interacciones de las entidades con los casos de uso. También, se diseña el sistema.

- Fase de construcción: se implementa el sistema *software* desde un punto de vista incremental, es decir, obteniendo cada vez versiones más completas y complejas basadas en las prioridades del cliente las cuales se extraen mayoritariamente en la fase anterior. También, se revisan y completan los casos de uso y se proponen cambios y mejoras a través de reuniones con el cliente.

- Fase de transición: el propósito de esta fase es asegurar que el sistema se encuentra en su etapa final de desarrollo ajustando errores y defectos localizados en las diferentes pruebas. Se elaboran los documentos finales (manual de instalación y configuración, manual de usuario, etc.) y se valida que el producto cumpla con las especificaciones. Por último, el sistema es desplegado en el entorno de producción.

La siguiente figura (Figura B.2) muestra un esquema general de AUP indicando las tareas y objetivos más comunes de cada una, sin olvidar los diferentes hitos del proyecto:

Figura B.2: Esquema general de fases en AUP.



Referencia bibliográfica: [81]

Cada una de las fases descritas anteriormente se puede dividir en iteraciones. Las iteraciones son una secuencia planificada de actividades (en cascada) ubicada dentro de una fase que finaliza en una versión (interna/externa).

B.2. Organización del proyecto. Estructura interna

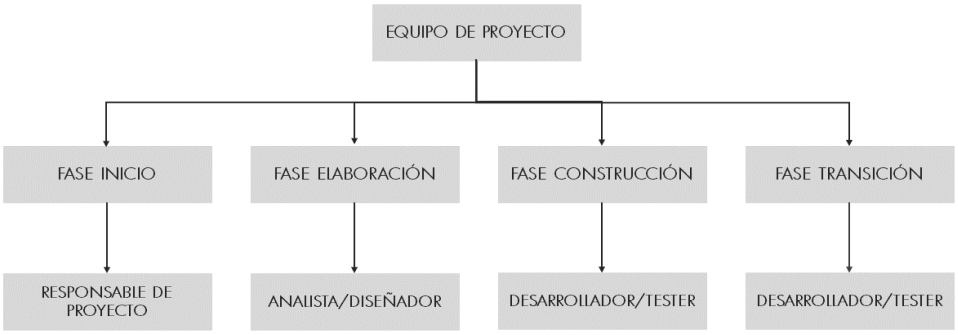
Otro aspecto importante en un proyecto es mostrar una visión superficial de la estructura interna de éste, referente a los roles de cada *stakeholder* participante en cada una de las fases indicadas (planificación, análisis y diseño, implementación y transición). Debido a que este proyecto ha sido realizado en el contexto de Trabajo Fin de Máster (TFM), todos los roles indicados a continuación han sido llevados a cabo por una única persona, el alumno **IGLESIAS GARCÍA, JESÚS**.

Sí es verdad que en todo proyecto colaborativo con un equipo de trabajo, en cada fase deberían existir diferentes roles desempeñados por diferentes personas. Sin embargo en el entorno presente,

al ser solamente una persona, ésta ha tenido que actuar solamente con el rol más relevante e importante de cada fase, razón por la que únicamente se muestra un rol en cada una de ellas.

La Figura B.3 muestra la estructura interna para el presente proyecto.

Figura B.3: Estructura interna.



De la imagen anterior, se pueden extraer los siguientes roles que forman la estructura interna del proyecto:

- Responsable del proyecto: graduado en Ingeniería Informática en Mención de Tecnologías de la Información (TI) y futuro graduado en Máster de Ingeniería Informática. Se encarga de ser el gerente de toda la asignación de tareas y recursos, supervisar el control, la planificación y el seguimiento del proyecto, es decir, mantener una correcta evolución del trabajo a lo largo de las iteraciones. Presenta conocimientos amplios en gestión de proyectos y en concreto en la metodología AUP.

- Analista de sistema: graduado en Ingeniería Informática en mención TI y futuro graduado en Máster de Ingeniería Informática. Se encarga de analizar el sistema y extraer las necesidades, funcionalidades y requerimientos del sistema requeridos por el cliente. Además, también valida los requisitos, casos de uso, etc. para garantizar que se cumple todo lo solicitado.

- Ingeniero *software*: también denominado diseñador. Graduado en Ingeniería Informática en mención TI y futuro graduado en Máster de Ingeniería Informática. Realiza las labores de diseñar el sistema a implementar con los diferentes modelos, presentando experiencia en el entorno de modelado *software*.

- Desarrollador: graduado en Ingeniería Informática en mención TI y futuro graduado en Máster de Ingeniería Informática. Se encarga de implementar y documentar el código fuente necesario para plasmar el diseño y la funcionalidad del sistema. También, ejecuta baterías

de prueba para asegurar el correcto funcionamiento del sistema. Presenta experiencia con la tecnología a utilizar.

- Tester: graduado en Ingeniería Informática en mención TI y futuro graduado en Máster de Ingeniería Informática. Se encarga de la medición y aseguramiento de la calidad de los procesos utilizados para crear un producto de calidad ajustándose a las necesidades específicas y a los planes establecidos anteriormente. Ejecuta exhaustivamente pruebas finales funcionales, de diseño y técnicas, todas ellas encaminadas a la detección de errores durante el proceso de desarrollo del producto.

B.3. Plan de fases

Todo proyecto debe comenzar con la elaboración de una planificación para poder tener una visión superficial de la distribución temporal y de recursos del mismo. Un buen artefacto que ayuda y facilita la planificación temporal es el plan de fases junto al plan de iteraciones. El plan de fases se elabora durante la fase de inicio de AUP y se corresponde con una planificación de alto nivel donde se puede observar claramente las diferentes fases de las que consta el proyecto así como el número de iteraciones en cada una de ellas, fechas estimadas de las mismas y objetivos propuestos.

Debido a que su elaboración se realiza al comienzo de un proyecto, se debe tener en cuenta que los períodos de tiempo indicados para cada fase son estimados. Estos períodos podrán ser modificados y/o sufrir retrasos por diversas dificultades y riesgos que puedan manifestarse a lo largo del desarrollo. Centrándonos en el plan de trabajo, la dimensión temporal de realización de este proyecto es de aproximadamente 9 meses (39 semanas) con una comunicación continuada del progreso del proyecto (aproximadamente cada 2-3 semanas) a mi tutor para permitir un seguimiento minucioso y una retroalimentación por parte de él.

Concluido el TFM, se puede afirmar que los intervalos de fechas expuestos han coincidido en una alta probabilidad con el trabajo real realizado, no presentando retrasos ni inconvenientes con respecto a la planificación inicial estimada y sí logrando un mínimo adelanto en la finalización.

A continuación, se muestra la distribución porcentual temporal y de esfuerzo estimada de cada fase y el plan de fases del proyecto:

	Inicio	Elaboración	Construcción	Transición
Esfuerzo	5 %	20 %	65 %	10 %
Tiempo	10 %	15,5 %	59 %	15,5 %

Tabla B.1: Estimación porcentual de cada fase

Fase	Iteraciones	Fecha Inicio	Fecha Final	Duración
Inicio	1	04/12/2017	31/12/2017	4 semanas
Elaboración	1	01/01/2018	11/02/2018	6 semanas
Construcción	2	12/02/2018	22/07/2018	23 semanas
Transición	1	23/07/2018	31/08/2018	6 semanas

Tabla B.2: Plan de fases del proyecto

El final de cada fase está marcado por un hito principal e hito secundario en el caso de final de iteración. En este proyecto, todo final de fase se corresponde con un final de iteración y a continuación se detalla a grandes rasgos el trabajo desarrollado en cada una de ellas permitiendo obtener una perspectiva del esquema seguido:

Fase	Hito
Inicio	Se ha estudiado el problema propuesto y la visión general de la solución del proyecto plasmando de esta manera la definición y alcance del proyecto, la planificación inicial, la estimación de costes y la gestión de riesgos. También, se analizó el estudio de arte de las tecnologías Blockchain (BC), profundizando en Hyperledger Fabric (HF). La finalización de esta fase corresponde al **hito 1**.
Elaboración	Una vez definido con mayor exactitud el proyecto en sí, se ha realizado todo el proceso de análisis y diseño del mismo para identificar los requisitos, actores, casos de uso, modelo de dominio, diseño de la arquitectura física del sistema, etc. La finalización de esta fase corresponde al **hito 2**.
Construcción	Dos iteraciones componen esta fase en la cual al final de cada una se produce una versión funcional e incremental del sistema. Durante la primera iteración se ha revisado y completado la definición de requisitos y casos de uso así como el diseño. En esta iteración[1], en primer lugar se preparó el entorno de trabajo y se definieron las diferentes partes de las que iba a constar el proyecto, implementando el componente de configuración de la Raspberry Pi (RPi), de monitorización de sucesos y de desencriptación. En la segunda iteración, se revisaron algunos errores que aparecieron a causa de la implementación durante la primera iteración y se desplegó la BC de HF definiendo la red de negocio y el sistema web. El **hito 3** y el **hito 4** se corresponden con el final de cada iteración, marcando también este último el final de fase.

[1]El anexo C muestra la gestión de tareas para la codificación del proyecto.

Transición	En esta última fase se han solucionado aquellos errores presentados en la segunda iteración (hito 4) de la fase de construcción y se ha llevado a cabo la ejecución de la batería de pruebas final. Además, se ha redactado la memoria, se han finalizado los manuales de usuario y de configuración del sistema y se ha preparado la versión definitiva del proyecto. La finalización de esta fase se corresponde con el **hito 5** donde se entrega el producto final junto con la documentación adicional.

Tabla B.3: Descripción de fases e hitos del proyecto

B.4. Plan de iteraciones

Indicado anteriormente, una iteración es una secuencia de actividades dentro de una fase con un plan establecido y unos criterios de evaluación, cuyo resultado es una versión interna/externa (hito secundario). Mientras que hito se puede definir como el punto de control en los cuales los participantes del proyecto revisan el progreso de éste.

En este apartado se desglosa cada una de las 5 iteraciones definidas, pudiendo observar a través de la planificación temporal tanto del calendario como del diagrama de Gantt los siguientes detalles: actividades englobadas, tiempo invertido en cada una (definido en días), recursos asignados y relaciones de precedencia las cuales condicionan el inicio en función de la predecesora.

En el momento de realizar la planificación inicial del proyecto para cada iteración se estableció un calendario laboral[2] compuesto por:

- **Días laborables**
  - Lunes-Viernes. Jornada laboral de 4 horas al día[3] con el siguiente horario:
    - De 17:00 a 21:00
  - Sábados. Jornada laboral de 4 horas al día con el siguiente horario:
    - De 10:00 a 14:00

[2]En la herramienta *Microsoft Project 2016* se ha creado un calendario con nombre "*calendar*" donde se ha definido la fecha de comienzo de cada iteración, el horario de trabajo, los días festivos y los recursos humanos involucrados en el proyecto junto con su coste por hora asociado.

[3]La dedicación al TFM ha sido parcial y no completa debido a que el recurso se encuentra en actividad laboral al momento de desarrollar este proyecto.

- **Días no laborables**
  - Domingo y festivos. Los festivos definidos han sido:
    - Día de la Constitución - 06/12/2017
    - Día de la Inmaculada - 08/12/2017
    - Natividad del Señor - 25/12/2017
    - Año Nuevo - 01/01/2018
    - Epifanía del Señor - 06/01/2018
    - Jueves Santo - 29/03/2018
    - Viernes Santo - 30/03/2018
    - Fiesta del Trabajo - 01/05/2018
    - Día de la Comunidad de Madrid - 02/05/2018
    - San Isidro - 15/05/2018
    - Asunción de la virgen - 15/08/2018

Para evitar al lector tener que instalar *software* adicional para la visión de la planificación de cada una de las cinco iteraciones, a continuación se muestran los calendarios y diagramas de Gantt:

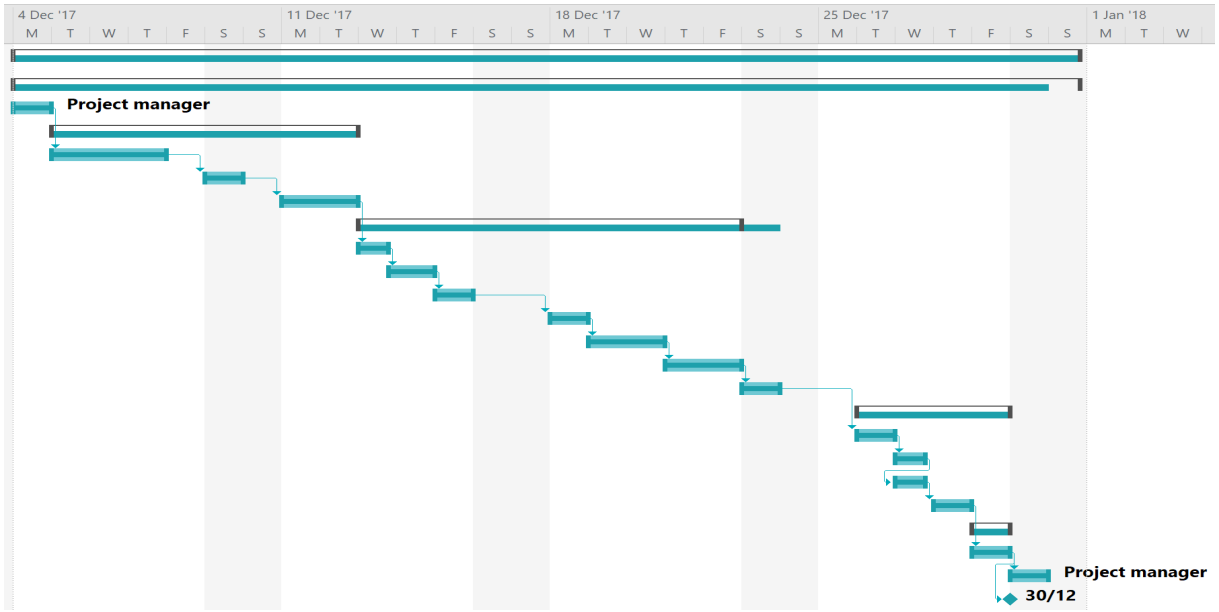
- **Iteración 1:** 24 días en total[4] (21 días laborables + 3 días festivos).

Figura B.4: Calendario de la fase de inicio - Iteración 1.

	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		Fase Inicio	24 days	Mon 4/12/17	Sun 31/12/17		
2		Iteración 1	24 days	Mon 4/12/17	Sun 31/12/17		
3		Planificación inicial	1 day	Mon 4/12/17	Mon 4/12/17		Project manager
4		Introducción y objetivos	5 days	Tue 5/12/17	Tue 12/12/17		Project manager
5		Definición, justificación y alcance del problema	2 days	Tue 5/12/17	Thu 7/12/17	3	
6		Objetivo general y material empleado	1 day	Sat 9/12/17	Sat 9/12/17	5	
7		Estudio de implementaciones similares	2 days	Mon 11/12/17	Tue 12/12/17	6	
8		Estudio del estado del arte	10 days	Wed 13/12/17	Sat 23/12/17		Project manager
9		Introducción	0,5 days	Wed 13/12/17	Wed 13/12/17	7	
10		Revisión del estado actual de la tecnología Blockchain	1,5 days	Wed 13/12/17	Thu 14/12/17	9	
11		Concepto de Blockchain	2 days	Fri 15/12/17	Sat 16/12/17	10	
12		Principales implementaciones de Blockchain	1 day	Mon 18/12/17	Mon 18/12/17	11	
13		Blockchain orientado al entorno IoT	2 days	Tue 19/12/17	Wed 20/12/17	12	
14		Hyperledger	2 days	Thu 21/12/17	Fri 22/12/17	13	
15		Lectura de proyectos relacionados	1 day	Sat 23/12/17	Sat 23/12/17	14	
16		Planificación del proyecto	4 days	Tue 26/12/17	Fri 29/12/17		Project manager
17		Organización del proyecto. Estructura interna	1 day	Tue 26/12/17	Tue 26/12/17	15	
18		Plan de fases	0,5 days	Wed 27/12/17	Wed 27/12/17	17	
19		Plan de iteraciones	0,5 days	Wed 27/12/17	Wed 27/12/17	18	
20		Estimación de costes	1 day	Thu 28/12/17	Thu 28/12/17	19	
21		Gestión de riesgos	1 day	Fri 29/12/17	Fri 29/12/17		Project manager
22		Identificación y análisis de riesgos	1 day	Fri 29/12/17	Fri 29/12/17	20	
23		Planificación inicial de la siguiente iteración	1 day	Sat 30/12/17	Sat 30/12/17	22	Project manager
24		Hito 1	0 days	Sat 30/12/17	Sat 30/12/17	22	Project manager

[4]Sin contar los días no laborables: Domingo.

Figura B.5: Diagrama de Gantt de la fase de inicio - Iteración 1.

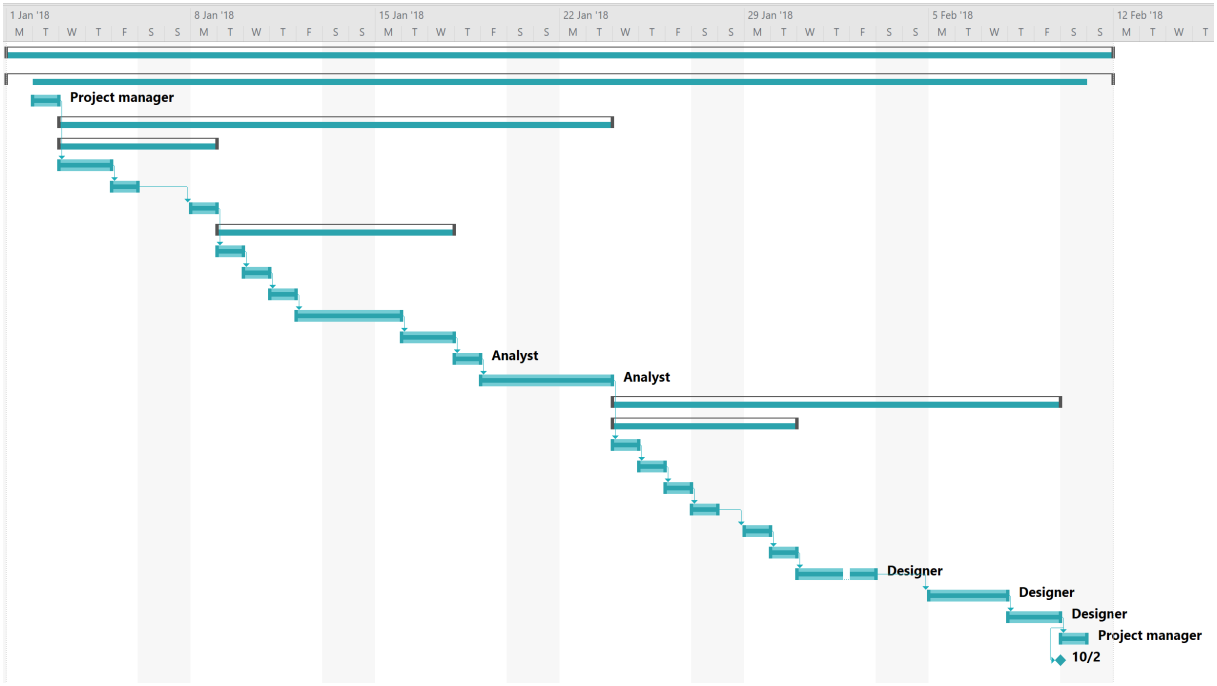


- **Iteración 2:** 36 días en total (34 días laborables + 2 días festivos).

Figura B.6: Calendario de la fase de elaboración - Iteración 2.

	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		Fase Elaboración	36 days	Mon 1/1/18	Sun 11/2/18		
2		Iteración 2	36 days	Mon 1/1/18	Sun 11/2/18		
3		Planificación inicial	1 day	Tue 2/1/18	Tue 2/1/18		Project manager
4		Análisis	17 days	Wed 3/1/18	Tue 23/1/18		
5		Requisitos	4 days	Wed 3/1/18	Mon 8/1/18		Analyst
6		Requisitos funcionales	2 days	Wed 3/1/18	Thu 4/1/18	3	
7		Requisitos no funcionales	1 day	Fri 5/1/18	Fri 5/1/18	6	
8		Requisitos de información	1 day	Mon 8/1/18	Mon 8/1/18	7	
9		Casos de uso	8 days	Tue 9/1/18	Wed 17/1/18		Analyst
10		Actores	1 day	Tue 9/1/18	Tue 9/1/18	8	
11		Lista de casos de uso	1 day	Wed 10/1/18	Wed 10/1/18	10	
12		Diagrama de casos de uso	1 day	Thu 11/1/18	Thu 11/1/18	11	
13		Descripción de casos de uso	3 days	Fri 12/1/18	Mon 15/1/18	12	
14		Escenarios	2 days	Tue 16/1/18	Wed 17/1/18	13	
15		Modelo de dominio	1 day	Thu 18/1/18	Thu 18/1/18	14	Analyst
16		Diagramas de secuencia de análisis	4 days	Fri 19/1/18	Tue 23/1/18	15	Analyst
17		Diseño software	15 days	Wed 24/1/18	Fri 9/2/18		
18		Modelado de la arquitectura física del sistema	6 days	Wed 24/1/18	Tue 30/1/18		Designer
19		Descripción de la arquitectura	1 day	Wed 24/1/18	Wed 24/1/18	16	
20		Diagrama de capas	1 day	Thu 25/1/18	Thu 25/1/18	19	
21		Diagrama de despliegue	1 day	Fri 26/1/18	Fri 26/1/18	20	
22		Diagrama de paquetes	1 day	Sat 27/1/18	Sat 27/1/18	21	
23		Diagrama de vista lógica	1 day	Mon 29/1/18	Mon 29/1/18	22	
24		Capa de persistencia	1 day	Tue 30/1/18	Tue 30/1/18	23	
25		Diagramas de secuencia de diseño	4 days	Wed 31/1/18	Sat 3/2/18	24	Designer
26		Diagrama de clases de diseño	3 days	Mon 5/2/18	Wed 7/2/18	25	Designer
27		Diagrama final de clases de diseño	2 days	Thu 8/2/18	Fri 9/2/18	26	Designer
28		Planificación inicial de la siguiente iteración	1 day	Sat 10/2/18	Sat 10/2/18	27	Project manager
29		Hito 2	0 days	Sat 10/2/18	Sat 10/2/18	27	Project manager

Figura B.7: Diagrama de Gantt de la fase de elaboración - Iteración 2.

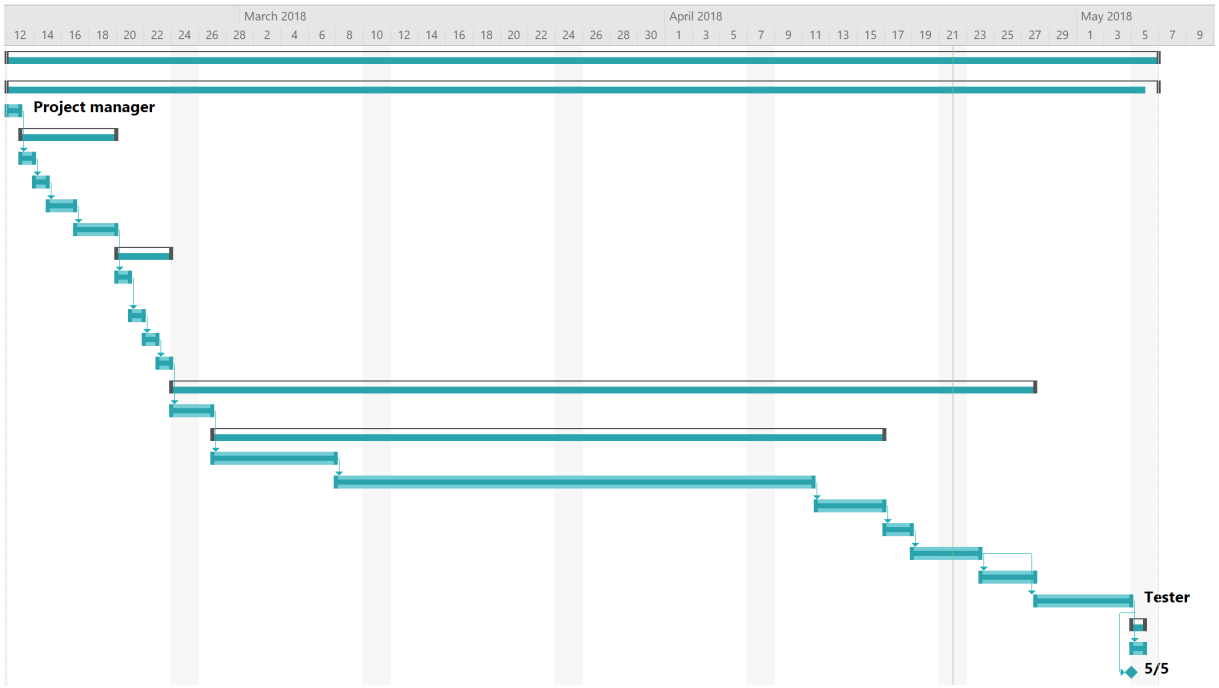


- **Iteración 3:** 72 días en total (68 días laborables + 4 días festivos).

Figura B.8: Calendario de la fase de construcción - Iteración 3.

	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		Fase Construcción	69 days	Mon 12/2/18	Sun 6/5/18		
2		Iteración 3	69 days	Mon 12/2/18	Sun 6/5/18		
3		Planificación inicial	1 day	Mon 12/2/18	Mon 12/2/18		Project manager
4		Revisión del análisis	6 days	Tue 13/2/18	Mon 19/2/18		Analyst
5		Revisión y completitud de requisitos	1 day	Tue 13/2/18	Tue 13/2/18	3	
6		Revisión y completitud del modelo de dominio	1 day	Wed 14/2/18	Wed 14/2/18	5	
7		Revisión y completitud de casos de uso	2 days	Thu 15/2/18	Fri 16/2/18	6	
8		Revisión y completitud de diagramas de secuencia de análisis	2 days	Sat 17/2/18	Mon 19/2/18	7	
9		Revisión del diseño software	4 days	Tue 20/2/18	Fri 23/2/18		Designer
10		Revisión y completitud del modelado de la arquitectura física del sistema	1 day	Tue 20/2/18	Tue 20/2/18	8	
11		Revisión y completitud de diagramas de secuencia de diseño	1 day	Wed 21/2/18	Wed 21/2/18	10	
12		Revisión y completitud de diagramas de clases de diseño	1 day	Thu 22/2/18	Thu 22/2/18	11	
13		Revisión y completitud del diagrama final de clases de diseño	1 day	Fri 23/2/18	Fri 23/2/18	12	
14		Desarrollo de la solución incremental	52 days	Sat 24/2/18	Fri 27/4/18		Developer
15		Preparación del entorno de trabajo	2 days	Sat 24/2/18	Mon 26/2/18	13	
16		Diseño y codificación de la prueba de concepto	40 days	Tue 27/2/18	Mon 16/4/18		
17		Componente de configuración del dispositivo Raspberry Pi	8 days	Tue 27/2/18	Wed 7/3/18	15	
18		Componente de monitorización de sucesos del entorno	27 days	Thu 8/3/18	Wed 11/4/18	17	
19		Componente de desencriptación de evidencia	4 days	Thu 12/4/18	Mon 16/4/18	18	
20		Integración y depuración del código	2 days	Tue 17/4/18	Wed 18/4/18	19	
21		Pruebas unitarias	4 days	Thu 19/4/18	Mon 23/4/18	20	
22		Pruebas de integración	4 days	Tue 24/4/18	Fri 27/4/18	21	
23		Ejecución de casos de prueba	4 days	Sat 28/4/18	Fri 4/5/18	21	Tester
24		Planificación inicial de la siguiente iteración	1 day	Sat 5/5/18	Sat 5/5/18		Project manager
25		Petición de cambios y/o modificaciones	1 day	Sat 5/5/18	Sat 5/5/18	23	
26		Hito 3	0 days	Sat 5/5/18	Sat 5/5/18	23	Project manager

Figura B.9: Diagrama de Gantt de la fase de construcción - Iteración 3.

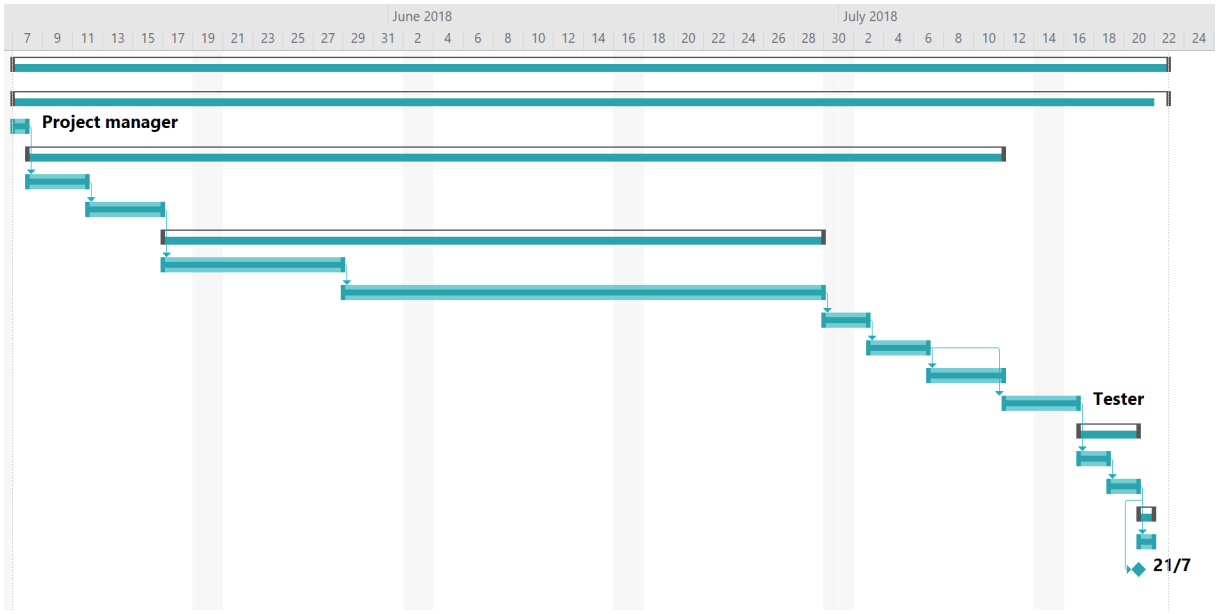


- **Iteración 4:** 66 días en total (65 días laborables + 1 día festivo).

Figura B.10: Calendario de la fase de construcción - Iteración 4.

	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		Fase Construcción	66 days	Mon 7/5/18	Sun 22/7/18		
2		Iteración 4	66 days	Mon 7/5/18	Sun 22/7/18		
3		Planificación inicial	1 day	Mon 7/5/18	Mon 7/5/18		Project manager
4		Desarrollo de la solución incremental	55 days	Tue 8/5/18	Wed 11/7/18		Developer
5		Resolución de errores de la iteracción anterior	4 days	Tue 8/5/18	Fri 11/5/18	3	
6		Introducción de cambios de la iteracción anterior	3 days	Sat 12/5/18	Wed 16/5/18	5	
7		Diseño y codificación de la prueba de concepto	38 days	Thu 17/5/18	Fri 29/6/18		
8		Protocolo de registro de incidencias en la Blockchain de Hyperledger Fabric	10 days	Thu 17/5/18	Mon 28/5/18	6	
9		Sistema web	28 days	Tue 29/5/18	Fri 29/6/18	8	
10		Integración y depuración del código	2 days	Sat 30/6/18	Mon 2/7/18	9	
11		Pruebas unitarias	4 days	Tue 3/7/18	Fri 6/7/18	10	
12		Pruebas de integración	4 days	Sat 7/7/18	Wed 11/7/18	11	
13		Ejecución de casos de prueba	4 days	Thu 12/7/18	Mon 16/7/18	11	Tester
14		Documentación	4 days	Tue 17/7/18	Fri 20/7/18		Developer
15		Manual de instalación y configuración , Versión inicial	2 days	Tue 17/7/18	Wed 18/7/18	13	
16		Manual de usuario , Versión inicial	2 days	Thu 19/7/18	Fri 20/7/18	15	
17		Planificación inicial de la siguiente iteración	1 day	Sat 21/7/18	Sat 21/7/18		Project manager
18		Petición de cambios	1 day	Sat 21/7/18	Sat 21/7/18	16	
19		Hito 4 , Release 1.0	0 days	Sat 21/7/18	Sat 21/7/18	16	Project manager

Figura B.11: Diagrama de Gantt de la fase de construcción - Iteración 4.

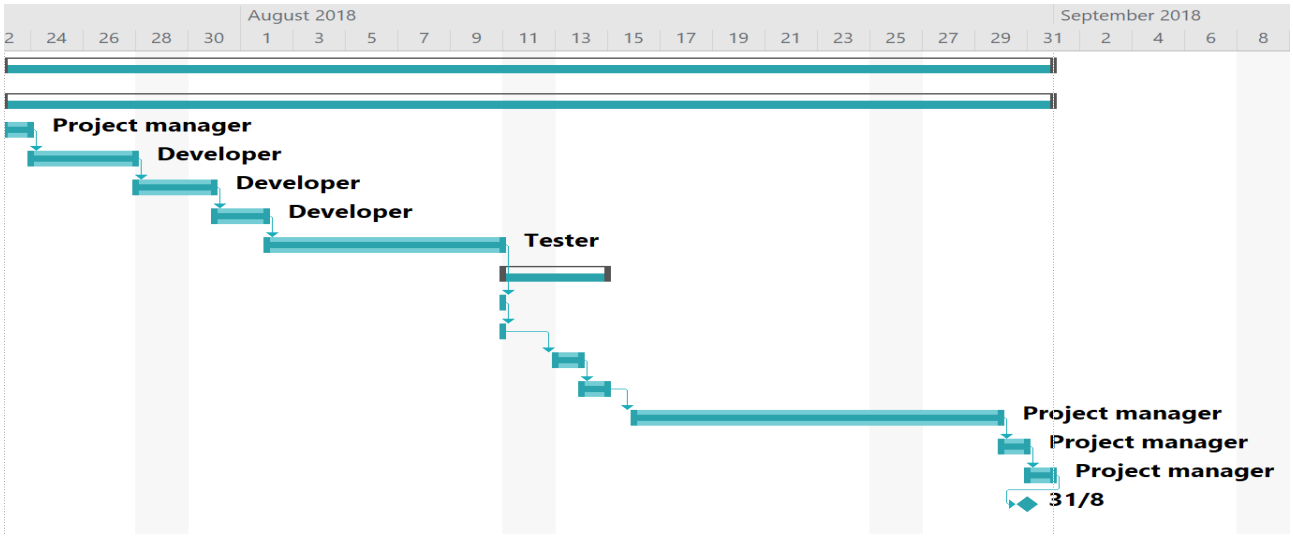


- **Iteración 5:** 35 días en total (34 días laborables + 1 día festivo).

Figura B.12: Calendario de la fase de transición - Iteración 5.

		Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1			Fase Transición	34 days	Mon 23/7/18	Fri 31/8/18		
2			Iteración 5	34 days	Mon 23/7/18	Fri 31/8/18		
3	✓		Planificación inicial	1 day	Mon 23/7/18	Mon 23/7/18		Project manager
4	✓		Resolución de errores de la iteracción anterior	4 days	Tue 24/7/18	Fri 27/7/18	3	Developer
5	✓		Introducción de cambios de la iteracción anterior	2 days	Sat 28/7/18	Mon 30/7/18	4	Developer
6	✓		Integración y depuración del código	2 days	Tue 31/7/18	Wed 1/8/18	5	Developer
7	✓		Ejecución final de casos de prueba	8 days	Thu 2/8/18	Fri 10/8/18	6	Tester
8			Documentación	3 days	Sat 11/8/18	Tue 14/8/18		Developer
9	✓		Prototipo Hyot. Información mostrada mediante el hardware	0,5 days	Sat 11/8/18	Sat 11/8/18	7	
10	✓		Esquema de pines GPIO	0,5 days	Sat 11/8/18	Sat 11/8/18	9	
11	✓		Manual de instalación y configuración	1 day	Mon 13/8/18	Mon 13/8/18	10	
12	✓		Manual de usuario	1 day	Tue 14/8/18	Tue 14/8/18	11	
13	✓		Redacción de la memoria	12 days	Thu 16/8/18	Wed 29/8/18	12	Project manager
14	✓		Preparación de la versión final	1 day	Thu 30/8/18	Thu 30/8/18	13	Project manager
15	✓		Despliegue de la versión final	1 day	Fri 31/8/18	Fri 31/8/18	14	Project manager
16	✓		Hito 5 , Entrega final	0 days	Fri 31/8/18	Fri 31/8/18	15	Project manager

Figura B.13: Diagrama de Gantt de la fase de transición - Iteración 5.



B.5. Estimación de costes

Otro aspecto importante a la hora de planificar y gestionar un proyecto es la elaboración de una estimación de costes totales que puede suponer el desarrollo de éste. En un proyecto real con uno o varios clientes y equipos de trabajo, la definición de estos costes debe aproximarse lo máximo posible a la realidad necesitada ya que puede conllevar al éxito o fracaso del proyecto. Estos costes a estimar incluyen tanto el coste de los recursos humanos involucrados (responsable del proyecto, equipo de desarrollo, equipo de aseguramiento de la calidad (*Quality Assurance* -QA-), analistas, diseñadores, etc.) como de herramientas *hardware* y *software* así como los costes de infraestructura, instalaciones y servicios consumibles.

Para realizar la estimación del coste de este proyecto se estimará como si se hubiera desarrollado profesionalmente a excepción de algunas salvedades citadas a continuación. En este caso se va a suponer que determinados dispositivos *hardware* han sido adquiridos para este proyecto con el fin de efectuar una estimación más realista. Los costes de infraestructura e instalación se obvian por el carácter de este trabajo: TFM. Con respecto a los servicios consumibles se tendrán en cuenta los cuantificables de una manera directa y obviando aquellos como por ejemplo: electricidad, material de oficina, etc. El desglose de costes es el siguiente:

Coste de recursos humanos			
Rol[5]	**Coste/hora (€)**	**Cantidad (días × horas)**	**Coste total (€)**
Responsable del proyecto	40 €	42 días × 4 horas	6720 €
Analista	35 €	23 días × 4 horas	3220 €
Diseñador	30 €	19 días × 4 horas	2280 €
Desarrollador	25 €	122 días × 4 horas	12200 €
Tester	25 €	16 días × 4 horas	1600 €
Coste total de recursos humanos			**26020 €**

Tabla B.4: Estimación del coste de recursos humanos

Coste de recursos humanos en cada fase			
Fase	**Recurso**	**Días laborales**	**Coste total (€)**
Inicio	Responsable del proyecto	21 días	3360 €
	Analista	-	-
	Diseñador	-	-
	Desarrollador	-	-
	Tester	-	-
			3360 €
Elaboración	Responsable del proyecto	2 días	320 €
	Analista	17 días	2380 €
	Diseñador	15 días	1800 €
	Desarrollador	-	-
	Tester	-	-
			4500 €
Construcción - 1ª iteración	Responsable del proyecto	2 días	320 €
	Analista	6 días	840 €
	Diseñador	4 días	480 €
	Desarrollador	52 días	5200 €
	Tester	4 días	400 €
			7240 €

[5]Cada rol puede presentar días laborales en varias iteraciones. Por ejemplo, el rol "Responsable del proyecto" presenta 21 días en la fase de inicio (1 iteración) y el resto de días pertenecen a las sucesivas iteraciones.

Construcción - 2ª iteración	Responsable del proyecto	2 días	320 €
	Analista	-	-
	Diseñador	-	-
	Desarrollador	59 días	5900 €
	Tester	4 dias	400 €
			6620 €
Transición	Responsable del proyecto	15 días	2400 €
	Analista	-	-
	Diseñador	-	-
	Desarrollador	11 días	1100 €
	Tester	8 días	800 €
			4300 €
Coste total de recursos humanos			**26020 €**

Tabla B.5: Estimación del coste de recursos humanos por fase

Coste de hardware			
Dispositivo	**Coste unitario (€)**	**Cantidad (Unidades)**	**Coste total (€)**
DELL U2713H	509 €	1ud	509 €
Macbook Pro Mid 2014	1000 €	1ud	1000 €
Magic Mouse 1	79 €	1ud	79 €
RPi 3 Starter Kit	151.37 €	1ud	151.37 €
RPi Cooling Kit	4.56 €	1ud	4.56 €
Kuman Starter Kit	39.99 €	1ud	39.99 €
RPi Camera Module V2 8MP	17.65 €	1ud	17.65 €
Breadboard, placa de expansión y cableado	8.96 €	1ud	8.96 €
LCD 16 × 2	3,95 €	1ud	3,95 €
Sensor HC-SR04	1.50 €	1ud	1.50 €
Coste total de *hardware*			**1815.98 €**

Tabla B.6: Estimación del coste de hardware

Coste de software			
Herramienta	**Coste unitario (€)**	**Cantidad (Unidades)**	**Coste total (€)**
Texpad	24.99 €	1ud	24.99 €
Resto de herramientas*[6]	0 €	1ud	0 €
Coste total de *software*			**24.99 €**

Tabla B.7: Estimación del coste de software

Coste de servicios consumibles			
Servicios	**Coste unitario (€)**	**Cantidad (Unidades)**	**Coste total (€)**
Conexión a Internet	40 €/mes x 7 meses	1ud	280 €
Coste total de servicios consumibles			**280 €**

Tabla B.8: Estimación del coste de servicios consumibles

Coste total del proyecto	
Apartado	**Coste total (€)**
Recursos humanos	26020 €
Hardware	1815.98 €
Software	24.99 €
Servicios consumibles	280 €
Total proyecto	**28140.97 €**

Tabla B.9: Estimación del coste total del proyecto

Por tanto, el presupuesto total estimado para la ejecución material del proyecto, asciende a la cantidad de 28140.97 €.

B.6. Gestión de riesgos

Se define *riesgo* como la probabilidad de que un problema potencial, circunstancia o eventualidad adversa al proyecto y/o al entorno de éste ocurra en el tiempo, imposibilitando la consecución de un objetivo y las consecuencias que puede conllevar si finalmente se materializa. La gestión de riesgos [44] por su parte, es la práctica de valorar y controlar los riesgos que afectan a un producto, proceso o proyecto *software* con el objetivo de poder anticiparse a los mismos antes de que éstos ocurran. La gestión del riesgo es necesaria debido a que:

[6]Todo el *software* utilizado ha sido empleando una licencia obtenida gratuitamente por ser estudiante, por ser *software* libre o bien se ha utilizado la versión gratuita.

- El riesgo del *software* es inherente en este ámbito.
- El riesgo aumenta a medida que aumenta la complejidad del sistema.
- El riesgo impide conseguir los objetivos si no se considera.
- El riesgo tiene su máximo valor al comienzo de un proyecto.
- El coste de reacción ante un riesgo es mayor a medida que el proyecto avanza.

Por todas las razones comentadas anteriormente, hoy en día se hace inviable no disponer de una gestión de riesgos. En la siguiente imagen (Figura B.14), se puede observar lo necesario y fundamental que es una buena gestión de riesgos desde el comienzo del proyecto con el fin de anticiparse a ellos y así asegurar con mayor probabilidad el éxito y calidad de un proyecto sin perjudicar gravemente en la planificación y/o en la estimación de costes.

Figura B.14: Relación coste-riesgo.

Referencia bibliográfica: [45]

B.6.1. Identificación y análisis de riesgos

El propósito de esta sección es, en primer lugar, identificar y realizar un análisis de los posibles riesgos que se puedan producir y en segundo lugar, establecer unas pautas de monitorización de los mismos por medio de la definición de un plan de acción para cada uno de ellos con el fin de que se puedan gestionar en caso de ocurrencia. Antes de listar los riesgos identificados, se expone la tipología principal de categorización que se ha seguido:

- Riesgo de proyecto: relacionado con las restricciones de recursos y relaciones externas necesarias para la realización del proyecto.
- Riesgo de proceso: relacionado con la consecución y revisión de cada una de las fases de desarrollo del proyecto.
- Riesgo de producto: relacionado con la experiencia en el dominio y con los resultados ineficientes en las fases previas al desarrollo.

Otro catalogación es aquella donde se clasifican en función de los siguientes dos conceptos:

- Impacto: mide el grado de fracaso y degradación del objetivo si se produce el riesgo.
- Probabilidad: mide la posibilidad de ocurrencia de un riesgo.

A continuación, se describen los principales riesgos detectados ordenados de mayor a menor según su exposición donde se considera **Riesgo = Probabilidad de amenaza × Magnitud de daño** y su valoración ha sido calculada según la norma estándar ISO/IEC 27005:2008[7] (Figura B.15).

Figura B.15: Estándar de gestión de riesgos ISO/IEC 27005:2008.

		Likelihood of Incident Scenario				
		Very Low	Low	Medium	High	Very High
Business Impact	Very Low	0	1	2	3	4
	Low	1	2	3	4	5
	Medium	2	3	4	5	6
	High	3	4	5	6	7
	Very High	4	5	6	7	8

Referencia bibliográfica: [76]

[7]Norma perteneciente a la familia ISO/IEC 27000. Consultar [46] para obtener más información.

Formulario de gestión de riesgos		
Identificador: 1	Fecha: 29/12/2017	Fase: En todas las fases
Título: **Planificación temporal errónea**		Categoría: Riesgo de proyecto
Impacto: Muy alto	Probabilidad: Media	Exposición: 6 (Fig. B.15)
Consecuencia: Retraso en el desarrollo del proyecto pudiendo derivar a una finalización tardía lo que supondría la cancelación del proyecto o insatisfacción por parte del cliente.		
Valoración de riesgos		
Descripción del riesgo: La planificación inicial no ha sido realizada de forma asumible y por tanto la duración de los plazos establecidos son irreales y no se pueden cumplir.		
Contexto del riesgo: Realizar durante la fase de inicio del proyecto una planificación inadecuada, ya sea por inexperiencia del responsable u otros motivos, puede provocar una asignación temporal completa que sea incorrecta y poco fiable.		
Análisis del riesgo: La planificación inicial es uno de los puntos donde se precisa de la máxima exactitud para alcanzar el éxito del proyecto. En la situación actual, existe una probabilidad media de que se produzca una planificación que posteriormente no se pueda cumplir. El impacto, en este caso es muy alto debido a que el proyecto debe estar finalizado antes de la fecha de entrega por lo que no se admite un posible retraso y es crucial realizar una estimación temporal lo más realista posible.		
Planificación de riesgos		
Estrategia: Protección del riesgo.		
Plan de contingencia preventivo: En la fase de inicio, tomar decisiones de planificación con la máxima exactitud a través del análisis en profundidad de los requerimientos del proyecto y ampliar los márgenes de error ante posibles fallos de planificación.		
Plan de contingencia mitigante: Efectuar una nueva planificación del proyecto restante de forma instantánea, asignando en el caso más extremo más carga de trabajo diaria para así poder cumplir los plazos establecidos.		
Punto de ruptura		
Si se alcanza un retraso mayor al 40 % sobre la planificación inicial, no existirá un método de recuperar el trabajo demorado ya que la única forma sería mediante la asignación de una carga mayor de trabajo lo cual no es asumible debido a la limitación de la existencia de un solo recurso para el desarrollo del proyecto. Por tanto, el proyecto no podrá ser entregado en el plazo establecido para ello.		
Resolución del riesgo		
Responsable: Jesús Iglesias García		
Rol: Responsable del proyecto		

Tabla B.10: Riesgo 1 - Planificación temporal errónea

Formulario de gestión de riesgos		
Identificador: 2	Fecha: 29/12/2017	Fase: Elaboración
Título: **Análisis del sistema inadecuado**		Categoría: Riesgo de proceso
Impacto: Alto	Probabilidad: Baja	Exposición: 4 (Fig. B.15)
Consecuencia: Sistema implementado sin satisfacer los requerimientos y funcionalidades previamente establecidas.		
Valoración de riesgos		
Descripción del riesgo: Gestión pobre durante la etapa de análisis del sistema, ya sea por una extracción de requisitos incorrecta o incompleta.		
Contexto del riesgo: Solamente un recurso humano a tiempo parcial se va a dedicar al análisis del sistema por lo que no es del todo posible establecer una organización y trabajo en equipo lo cual proporcionaría en gran medida un mejor análisis. Es por ello que este recurso es el encargado de la decisión y si no posee demasiada experiencia o el proyecto no se encuentra bien definido, puede proporcionar un planteamiento erróneo.		
Análisis del riesgo: Cuando se inicia un nuevo proyecto, la asignación de recursos humanos se hace en base a obtener el máximo rendimiento y calidad final del producto/servicio por lo que se asignan en la medida de lo posible recursos con experiencia en cada fase. En este caso debido a la naturaleza del proyecto, se ha asignado a un analista a tiempo parcial para efectuar el análisis de un sistema que se encuentra bien definido por lo que la probabilidad de riesgo es baja y su impacto se considera alto ya que afectaría de forma negativa al desarrollo normal del proyecto, repercutiendo a su vez en una posible reestructuración del proyecto entero y por tanto afectando a la planificación.		
Planificación de riesgos		
Estrategia: Protección del riesgo.		
Plan de contingencia preventivo: Concretar revisiones de los requisitos para refinarlos durante la primera iteración de la fase de construcción.		
Plan de contingencia mitigante: Mejorar la definición del proyecto y analizar detenidamente el sistema de nuevo cuando surjan los primeros indicios de un desarrollo incompleto o incorrecto.		
Punto de ruptura		
Si durante la mitad de la segunda iteración de la fase de construcción, los requisitos y funcionalidades se manifiestan como incompletos o erróneos y suponen un 30 % del total del sistema a construir, el proyecto se considera como no asumible al no cumplir con los plazos establecidos.		
Resolución del riesgo		
Responsable: Jesús Iglesias García		
Rol: Analista del proyecto		

Tabla B.11: Riesgo 2 - Análisis del sistema inadecuado

Formulario de gestión de riesgos		
Identificador: 3	Fecha: 29/12/2017	Fase: Elaboración
Título: **Diseño del sistema inadecuado**		Categoría: Riesgo de proceso
Impacto: Alto	Probabilidad: Baja	Exposición: 4 (Fig. B.15)
Consecuencia: Sistema implementado sin satisfacer la arquitectura física previamente establecida.		
Valoración de riesgos		
Descripción del riesgo: Gestión pobre durante la etapa de diseño del sistema, ya sea por un modelado de la arquitectura física del sistema incorrecta o incompleta.		
Contexto del riesgo: Solamente un recurso humano a tiempo parcial se va a dedicar al diseño por lo que es esencial que conozca profundamente la definición del proyecto y el análisis previamente realizado sobre éste y en base a estos dos aspectos encargarse de la decisión. Por lo que si no posee demasiada experiencia o el proyecto no se encuentra bien definido, puede plantear una arquitectura errónea.		
Análisis del riesgo: En este caso debido a la naturaleza del proyecto, tanto éste como su análisis se encuentran bien definidos por lo que la probabilidad de un diseño erróneo es baja. Su impacto se considera alto ya que afectaría de forma negativa al desarrollo normal del proyecto, repercutiendo a su vez en una posible reestructuración del proyecto entero y por tanto afectando a la planificación.		
Planificación de riesgos		
Estrategia: Protección del riesgo.		
Plan de contingencia preventivo: Concretar varias revisiones del modelado físico de la arquitectura al comienzo de la primera iteración de la fase de construcción.		
Plan de contingencia mitigante: Analizar la arquitectura actual y las posibles variantes a adoptar cuando surjan los primeros indicios de un diseño inestable.		
Punto de ruptura		
Si al comienzo de la segunda iteración de la fase de construcción, el modelado de la arquitectura física se considera incompleto o erróneo y supone una reestructuración de al menos el 50 % del sistema a construir, el proyecto se considera como no asumible al no cumplir con los plazos establecidos.		
Resolución del riesgo		
Responsable: Jesús Iglesias García		
Rol: Diseñador del proyecto		

Tabla B.12: Riesgo 3 - Diseño del sistema inadecuado

Formulario de gestión de riesgos		
Identificador: 4	Fecha: 29/12/2017	Fase: Construcción y transición
Título: **Inexperiencia con la tecnología**		Categoría: Riesgo de producto
Impacto: Alto	Probabilidad: Baja	Exposición: 4 (Fig. B.15)
Consecuencia: El desconocimiento y falta de experiencia de la tecnología de desarrollo podría provocar un retraso en la planificación al tener que obtener formación o una implementación incorrecta o ineficiente.		
Valoración de riesgos		
Descripción del riesgo: Falta de conocimiento de los lenguajes de programación, *frameworks* y librerías a utilizar para el desarrollo.		
Contexto del riesgo: El desarrollador presenta experiencia en alguna de las tecnologías a utilizar mientras que en otras carece de ella. Si bien es cierto que existe documentación exhaustiva donde apoyarse, no se descarta que esta documentación en algunos casos sea escasa debido a que se tratan de tecnologías nuevas lo que podría convertirse en un contratiempo.		
Análisis del riesgo: Para desarrollar el proyecto se dispone en principio del conocimiento necesario para abordarlo sin ningún inconveniente sumado a la documentación detallada existente y con ejemplos. En principio, la probabilidad de ocurrencia de este riesgo es baja. Sin embargo, en caso de ocurrir se consideraría un grave problema ya que afectaría de manera directa a la realización del proyecto causando posibles retrasos siendo por este motivo su impacto alto.		
Planificación de riesgos		
Estrategia principal: Protección del riesgo.		
Plan de contingencia preventivo: Consultar referencias sobre aquellas tecnologías donde se presenten dudas, apoyándose en especial en la documentación oficial y sus ejemplos ya que facilitan el entendimiento de la estructura y funcionalidad del código a través de la exposición de la interfaz de programación (*Application Programming Interface* -API-).		
Plan de contingencia mitigante: Establecer durante la primera iteración de la fase de desarrollo un marco temporal de aprendizaje sobre las tecnologías a utilizar. Esto sería posible gracias a la holgura existente en la planificación inicial.		
Punto de ruptura		
Si el desarrollador presenta problemas a la hora de codificar el sistema, no considerándose completamente funcional y produciéndose un retraso del 40 % con respecto a la planificación inicial, el proyecto se considera como no subsanable debido a la falta de tiempo y recursos para solucionar los errores presentados.		
Resolución del riesgo		
Responsable: Jesús Iglesias García		
Rol: Desarrollador del proyecto		

Tabla B.13: Riesgo 4 - Inexperiencia con la tecnología

Formulario de gestión de riesgos		
Identificador: 5	Fecha: 29/12/2017	Fase: En todas las fases
Título: **Ausencia de personal**		Categoría: Riesgo de proyecto
Impacto: Muy alto	Probabilidad: Muy baja	Exposición: 4 (Fig. B.15)
Consecuencia: Retraso en la finalización del proyecto, no cumpliendo por tanto los plazos planificados inicialmente.		
Valoración de riesgos		
Descripción del riesgo: Imposibilidad de avanzar en el proyecto por causas de fuerza mayor ya sea por motivos internos o externos al proyecto, repercutiendo en el retraso de las entregas o en el aumento de la cantidad de trabajo para finalizar en el mismo instante de tiempo.		
Contexto del riesgo: Está presente durante todo el ciclo de vida del proyecto ya que al ser un factor humano puede manifestarse en cualquier instante.		
Análisis del riesgo: Este riesgo ha sido marcado con un valor de probabilidad muy bajo ya que aunque depende de factores variables y se puede dar en cualquier instante, no es una situación habitual que afecte de manera permanente al desarrollo del proyecto. Si que es verdad que posee un impacto muy alto ya que la ausencia continuada y en gran cantidad repercute negativamente al avance.		
Planificación de riesgos		
Estrategia principal: Reducción del riesgo		
Plan de contingencia preventivo: Establecer márgenes temporales en la planificación para afrontar estos posibles sucesos de tal manera que no repercutan o que repercutan lo mínimo posible. Replanificar la carga de trabajo restante para finalizar en el mismo plazo.		
Plan de contingencia mitigante: Replanificar la carga de trabajo y el marco temporal restante para adaptarse a los requisitos mínimos que debe contener la solución a implementar de forma que se incluyan más funcionalidades progresivamente si da tiempo.		
Punto de ruptura		
Si la ausencia de personal es elevada provocando severos retrasos con respecto a la planificación (40 %), la situación es irremediable y no se puede afrontar un reparto de la carga de trabajo restante ya que se consideraría excesivo. Debido a la naturaleza del proyecto, tampoco se dispone de otros recursos humanos que puedan continuar con el proyecto.		
Resolución del riesgo		
Responsable: Jesús Iglesias García		
Rol: Responsable del proyecto		

Tabla B.14: Riesgo 5 - Ausencia de personal

Formulario de gestión de riesgos		
Identificador: 6	Fecha: 29/12/2017	Fase: Construcción y transición
Título: **Bugs en implementación**		Categoría: Riesgo de producto
Impacto: Medio	Probabilidad: Baja	Exposición: 3 (Fig. B.15)
Consecuencia: Desarrollar un programa informático que presente errores de funcionalidad, diseño, seguridad, etc. supone que el tester notifique estos problemas al desarrollador y éste tenga que emplear tiempo y esfuerzo para su resolución lo cual a su vez implica la replanificación.		
Valoración de riesgos		
Descripción del riesgo: Una elevada complejidad algorítmica y de la solución puede suponer dificultad a la hora de codificar y por tanto desarrollar un sistema con funcionalidad incorrecta que no cumpla las necesidades requeridas.		
Contexto del riesgo: El desarrollador debido al desconocimiento de las tecnologías puede cometer errores en la codificación o incluso estar debido a una especificación incorrecta de la solución.		
Análisis del riesgo: Aunque en principio el análisis y diseño del proyecto están correctamente definidos puede que durante el transcurso del ciclo de vida de éste se realice alguna modificación que el desarrollador olvide implementar. Incluso puede ser el desarrollador el que rompa alguna funcionalidad anterior al incluir nuevas. Aun así se considera que tanto la probabilidad como su impacto sean medios debido a una detección temprana por el tester. Esto implicaría una notificación prematura, reduciendo así el riesgo con un muy leve retraso en la codificación de alguna tarea específica subsanable con el aumento de la carga en otro día laboral.		
Planificación de riesgos		
Estrategia principal: Prevención del riesgo.		
Plan de contingencia preventivo: Al finalizar cada iteración de la fase de desarrollo, efectuar una batería de pruebas tanto funcionales como técnicas con el fin de detectar lo antes posible cualquier error en la solución.		
Plan de contingencia mitigante: Efectuar al finalizar la introducción de una nueva funcionalidad una batería de pruebas exhaustiva con lo que la detección de cualquier problema sería en su fase más temprana y su resolución no supondría un contratiempo grande en la planificación ni un retraso en el proyecto.		
Punto de ruptura		
Si se localizan demasiados fallos de funcionalidad y diseño durante la fase de transición y esta cifra alcanza un 40 % se considera que el tiempo necesario para revertir esta situación sobrepasaría la fecha límite de entrega o tendría un sobrecoste inasumible.		
Resolución del riesgo		
Responsable: Jesús Iglesias García		
Rol: Desarrollador y tester del proyecto		

Tabla B.15: Riesgo 6 - Bugs en la implementación

Formulario de gestión de riesgos		
Identificador: 7	Fecha: 29/12/2017	Fase: En todas las fases
Título: **Carencia de presupuesto**		Categoría: Riesgo de proyecto
Impacto: Medio	Probabilidad: Muy baja	Exposición: 2 (Fig. B.15)
Consecuencia: No poder afrontar las actuales o nuevas necesidades del proyecto debido a la adquisición de recursos materiales provocaría retrasos en la planificación o no poder implementar todas las funcionalidades requeridas.		
Valoración de riesgos		
Descripción del riesgo: Aun habiendo adquirido todo el material necesario, puede existir en cualquier momento la inclusión de una nueva funcionalidad que necesite de un nuevo recurso material lo cual provoca costes económicos adicionales y pueden suponer no disponerlos, derivando en incompletitud de las funcionalidades del sistema.		
Contexto del riesgo: Está presente durante todo el ciclo de vida del proyecto ya que constantemente se pueden requerir cambios que supongan costes adicionales a la estimación realizada en la planificación inicial.		
Análisis del riesgo: La correcta definición del proyecto y por tanto también de los recursos materiales necesitados provocan que en principio no se necesite nada a mayores por lo que se determina que la probabilidad de este riesgo es muy baja y el impacto medio ya que podría darse el caso de no poder implementar una nueva funcionalidad al no poder adquirir el recurso material.		
Planificación de riesgos		
Estrategia principal: Protección del riesgo		
Plan de contingencia preventivo: Realizar durante la fase de inicio una estimación del coste global lo más aproximada posible, estudiando el proyecto con profundidad para conocer los recursos necesarios y así poder adaptarse al presupuesto existente.		
Plan de contingencia mitigante: Reorganizar los recursos existentes para reducir en la medida de lo posible los inconvenientes que se puedan producir.		
Punto de ruptura		
Si la estimación de coste del nuevo material a adquirir supera el disponible no se puede aceptar dicha petición y por tanto no se puede implementar lo acordado.		
Resolución del riesgo		
Responsable: Jesús Iglesias García		
Rol: Responsable del proyecto		

Tabla B.16: Riesgo 7 - Carencia de presupuesto

Formulario de gestión de riesgos		
Identificador: 8	Fecha: 29/12/2017	Fase: En todas las fases
Título: **Fallo de hardware**		Categoría: Riesgo de producto
Impacto: Muy bajo	Probabilidad: Baja	Exposición: 1 (Fig. B.15)
Consecuencia: Retraso en las tareas del proyecto debido a la no disponibilidad de los recursos físicos necesarios para avanzar. También se pueden producir pérdidas de información importante.		
Valoración de riesgos		
Descripción del riesgo: Los dispositivos *hardware* pueden averiarse o extraviarse durante el proyecto lo que afectaría al desarrollo normal de éste.		
Contexto del riesgo: Está presente durante todo el proceso de desarrollo del proyecto ya que el mal funcionamiento o deterioro *hardware* no es controlable directamente.		
Análisis del riesgo: Este riesgo se ha marcado con probabilidad baja debido a que el *hardware* cada vez se construye con componentes más fiables y duraderos que no tienden al fallo. El impacto sería muy bajo ya que es muy común disponer de otros elementos *hardware* como segunda opción siempre y cuando éstos no presenten un coste elevado. Para el proyecto actual, el *hardware* necesitado es básico lo cual no implicaría un inconveniente a considerar si se produjese el fallo en alguno de ellos.		
Planificación de riesgos		
Estrategia principal: Aceptación del riesgo.		
Plan de contingencia preventivo: Realizar revisiones periódicas del estado de los sistemas *hardware* y mantener estos sistemas en buena disposición.		
Plan de contingencia mitigante: Disponer de dispositivos alternativos y copias de seguridad de la información importante para poder continuar el desarrollo del proyecto sin afectar gravemente a la planificación.		
Punto de ruptura		
Si el desarrollador durante la segunda iteración de la fase de construcción experimenta un fallo de *hardware* perdiendo el código fuente del sistema por completo, se asume como no recuperable en el tiempo de planificación restante.		
Resolución del riesgo		
Responsable: Jesús Iglesias García		
Rol: Responsable del proyecto		

Tabla B.17: Riesgo 8 - Fallo de hardware

Formulario de gestión de riesgos		
Identificador: 9	Fecha: 29/12/2017	Fase: En todas las fases
Título: **Fallo de software**		Categoría: Riesgo de producto
Impacto: Muy bajo	Probabilidad: Baja	Exposición: 1 (Fig. B.15)
Consecuencia: Retraso en el avance del proyecto y/o pérdidas de información importante.		
Valoración de riesgos		
Descripción del riesgo: Funcionamiento incorrecto del *software* debido a problemas del propio *software* como por ejemplo renovación de licencias o a problemas externos como amenazas en forma de virus. Esto provocaría incidencias puntuales o duraderas que afectarían al desarrollo normal de la planificación e incluso a posibles pérdidas de datos e información.		
Contexto del riesgo: Se produce en todas las fases del proyecto debido a que constantemente se utiliza *software* para desarrollar el proyecto.		
Análisis del riesgo: Este riesgo se ha marcado con probabilidad baja debido a que hoy en día el *software* es fiable y su tasa de fallo es reducida. En caso de producirse, existen herramientas similares que permiten continuar la misma tarea sin afectar drásticamente por lo que el impacto provocado sería mínimo.		
Planificación de riesgos		
Estrategia principal: Reducción del riesgo.		
Plan de contingencia preventivo: Mantener el *software* actualizado y al día en cuanto a licencias junto con la realización periódica de copias de seguridad para prevenir la pérdida de información.		
Plan de contingencia mitigante: Disponibilidad de herramientas similares para su utilización que permitan realizar las tareas afectadas y restaurar información a través de copias de seguridad.		
Punto de ruptura		
Si el desarrollador a mitad de la fase de construcción experimenta un fallo de *software* perdiendo el código fuente del sistema por completo, se asume como no recuperable en el tiempo de planificación restante.		
Resolución del riesgo		
Responsable: Jesús Iglesias García		
Rol: Responsable del proyecto		

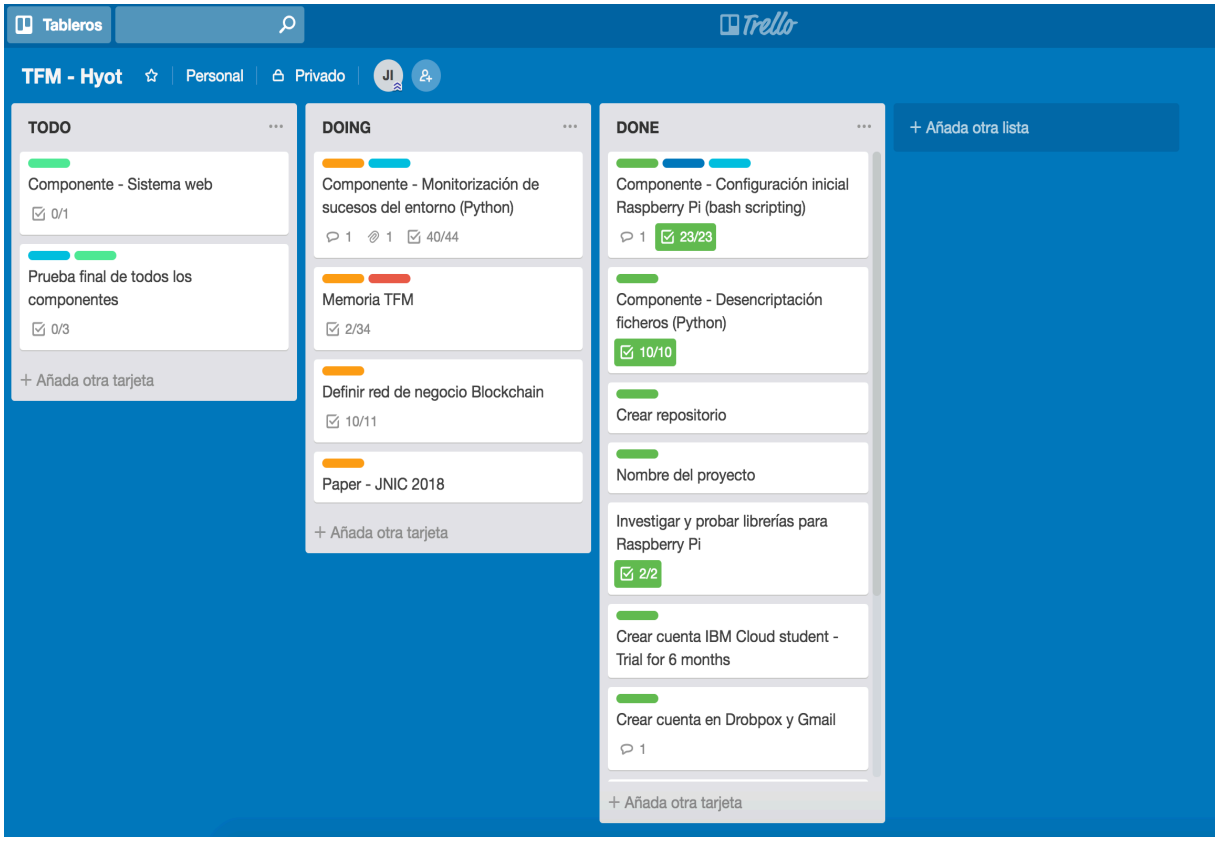
Tabla B.18: Riesgo 9 - Fallo de software

Anexo C

Planificación de tareas con Trello

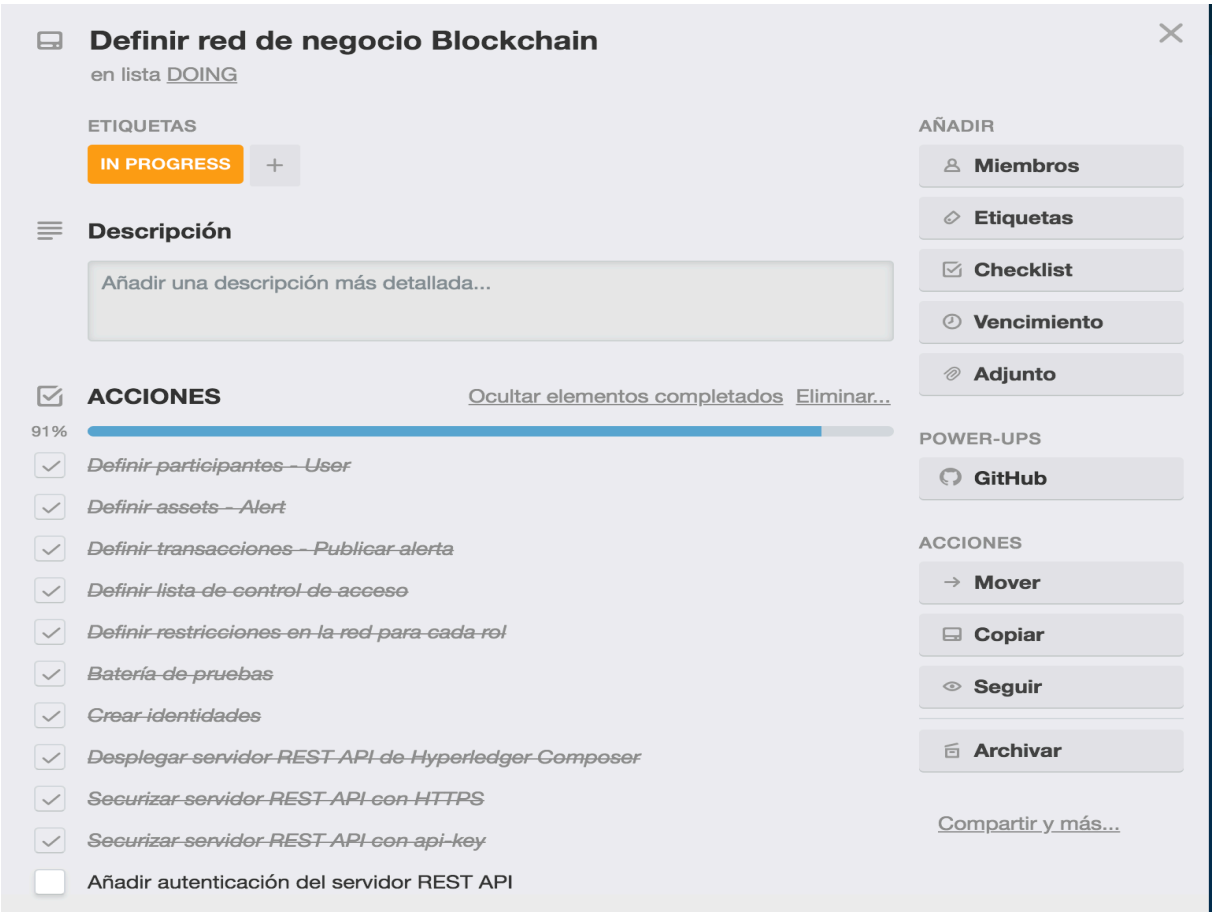
Para la organización del proyecto y para mantener un registro del avance se ha empleado la herramienta Trello con el fin de establecer una planificación de tareas -estilo Kanban- tal y como muestra la Figura C.1 en un instante determinado del progreso. En ella, se distinguen las tareas principales clasificándolas en tres posibles grupos según estén pendiente de realizar, estén en progreso o se encuentren ya finalizadas.

Figura C.1: Planificación de tareas con Trello.



A su vez cada tarea principal puede desglosarse en subtareas como muestra la Figura C.2 que hace referencia a la definición de la red de negocio.

Figura C.2: Detalle de una tarea en Trello.

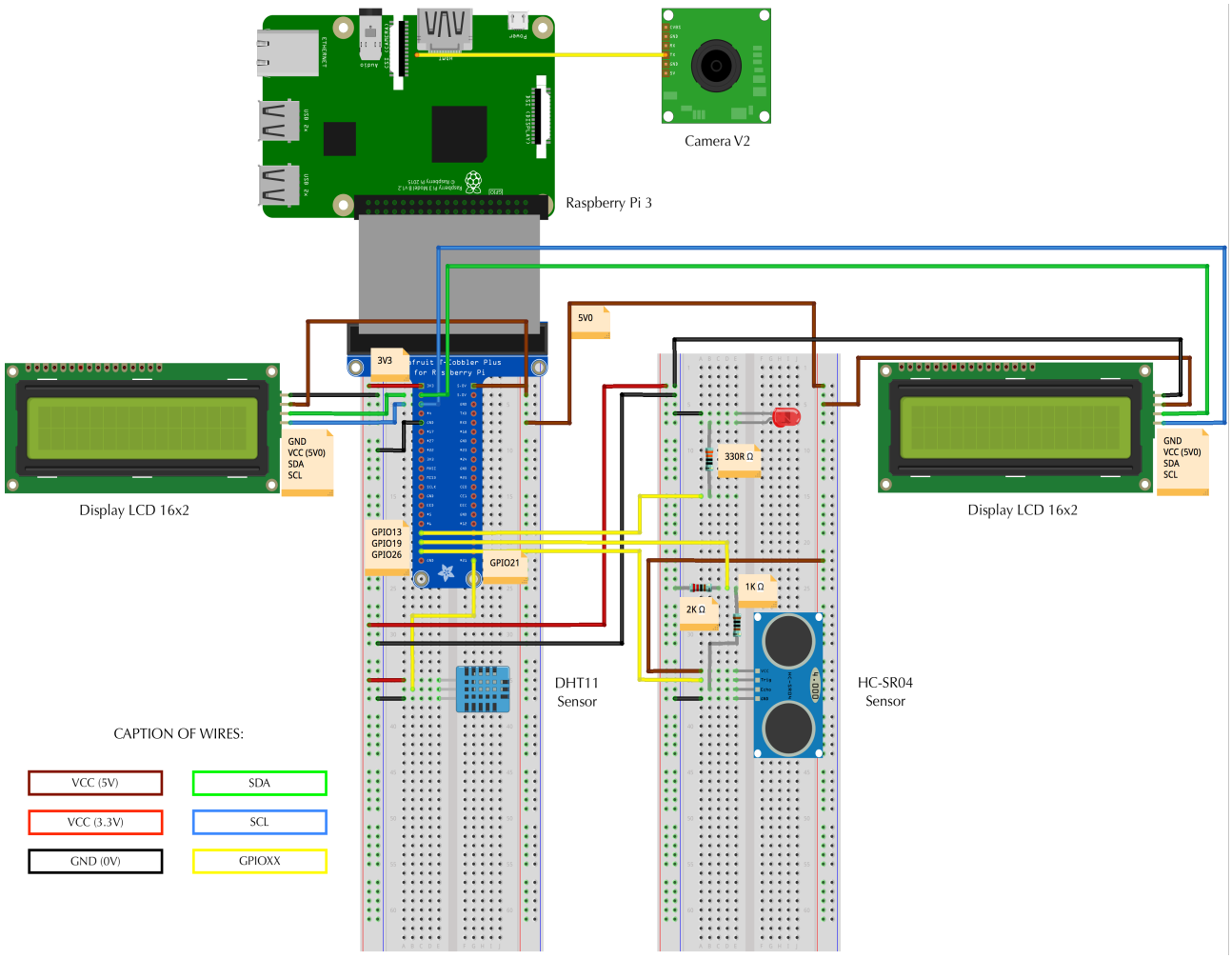


Anexo D

Prototipo de Hyot

La Figura D.1 muestra el esquema del prototipo de Hyot utilizado por el componente de monitorización de sucesos del entorno.

Figura D.1: Esquema del prototipo de Hyot.



Este prototipo está compuesto de diferentes dispositivos electrónicos donde es importante cerciorarse de una correcta conexión de todos ellos para el adecuado funcionamiento. Entre ellos:

- Raspberry Pi[1] (RPi), conectada a la fuente eléctrica y a Internet ya sea a través del estándar ethernet o WiFi. Adicionalmente, puede requerir la conexión de los periféricos teclado y ratón a través de los puertos USB existentes en la placa.

- Módulo de cámara V2, conectada directamente a la interfaz serie para cámaras (*Camera Serial Interface* -CSI-) de la RPi.

- *Breadboard* o placa de pruebas (2), para la conexión de los diferentes componentes.

- Placa de extensión tipo T (T-Cobbler) de los pines de propósito general de entrada/salida (*General Purpose Input Output* -GPIO-) conectada a un cable plano y a la placa de pruebas.

- Cable plano de 40 pines, para la conexión entre el módulo GPIO de la RPi y la placa de extensión tipo T.

- Pantalla LCD (*Liquid Crystal Display*) 16x2 (2), dispositivos de salida conectados mediante el protocolo I2C (*Inter-Integrated Circuit*) a los pines correspondientes del módulo GPIO. Se debe tener en cuenta que la alimentación de estas pantallas debe ser de 5 voltios (5V).

- Sensor DHT-11, para monitorizar la temperatura y humedad. Este dispositivo de entrada es conectado a VCC (*Voltage at the Common Collector*) con 3.3V, a tierra (*Ground* -GND-) y al pin GPIO 21. No es necesario el montaje de una resistencia debido a que en la placa de circuito impreso del propio sensor incluye una resistencia de 10000 ohmios (10KΩ)[2]

- Sensor HC-SR04, para monitorizar la distancia. Este dispositivo de entrada es conectado a VCC (5V) ya que funciona con esta tensión eléctrica, a tierra (GND), al pin GPIO 26 el pin de disparo (*Trigger*) y para conectar el pin *Echo* al pin GPIO 19 hay que utilizar un divisor de voltaje con el fin de reducir la tensión eléctrica de salida del sensor (5V) al necesitado a recibir por el módulo GPIO de la RPi (3.3V), evitando de esta forma cualquier acción dañina. Para ello, dicho pin se conecta a un extremo de una resistencia de 1KΩ. El otro extremo de ésta se conecta al pin GPIO 19 que a su vez está conectado en la misma línea donde se coloca otra resistencia de 2KΩ. Esta última está conexionada directamente a GND.

- LED (*Light-Emitting Diode*) de color rojo, dispositivo de salida donde el cátodo (terminal más corto) está conectado a GND y el ánodo (terminal más largo) a un extremo de una

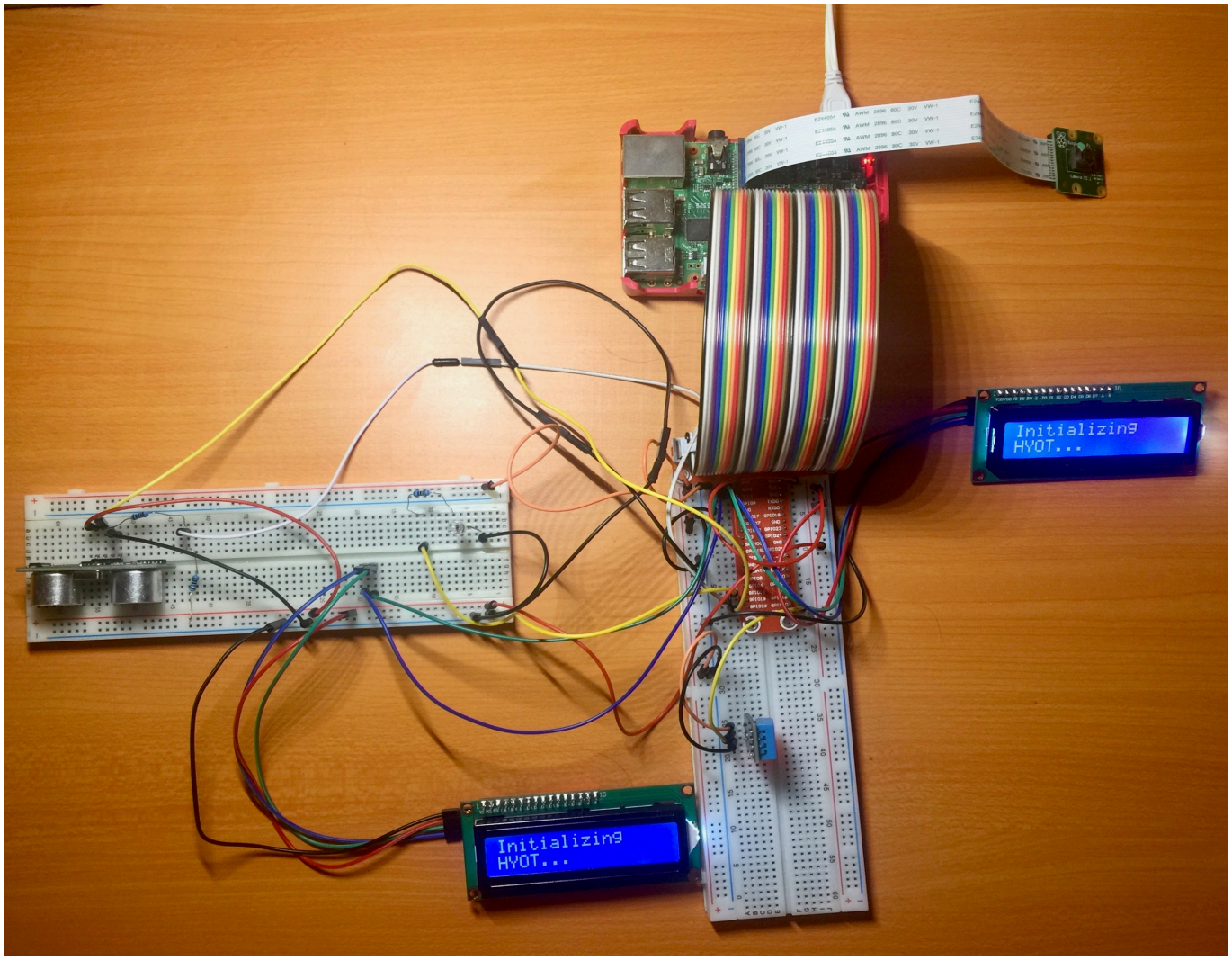
[1]Para el proyecto Hyot se ha empleado la versión 3.

[2]En la referencia [1] se explica el código de colores de las resistencias eléctricas.

resistencia[3] de 330Ω[4]. El otro extremo de ésta está conectado al pin GPIO 13.

La Figura D.2 muestra el prototipo *hardware* real de Hyot el cual contiene los mismos componentes que los citados previamente.

Figura D.2: Prototipo hardware real de Hyot.



[3]La conexión de una resistencia sirve para limitar el paso de la intensidad de corriente eléctrica a través del circuito lo que protege tanto al propio LED como al pin de la RPi. De forma que cuanta más resistencia eléctrica, más se atenuará el LED hasta el punto de anular este paso de corriente que implicará la completa atenuación.

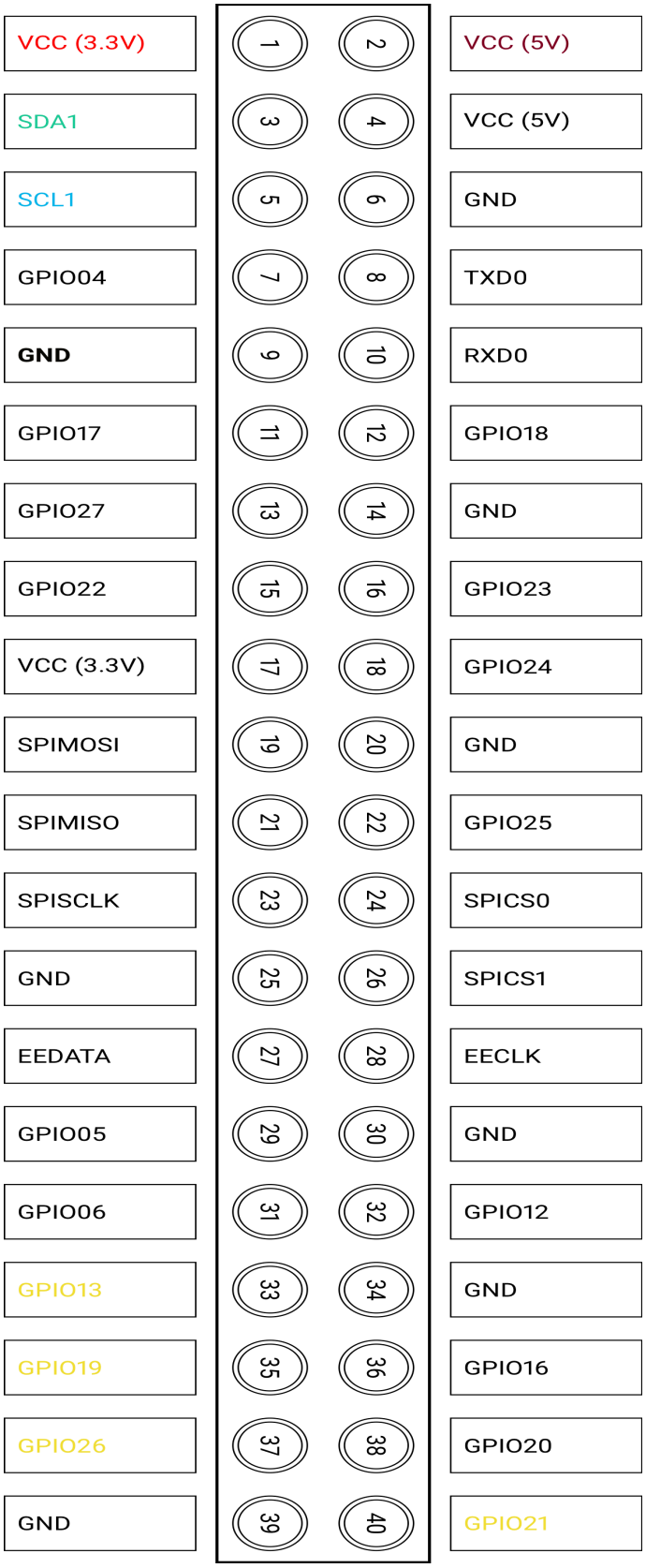
[4]El valor necesario para la resistencia puede ser calculada con la ley de Ohm.

Anexo E

Esquema de pines GPIO empleados en Hyot

La Figura E.1 muestra el esquema de los pines GPIO (*General Purpose Input Output*) de la Raspberry Pi (RPi) en modo BCM [74] empleados en el prototipo de Hyot. Cabe decir, que se puede utilizar en todo momento otros pines siempre y cuando presenten la misma funcionalidad y/o se configure en el código de los componentes.

Figura E.1: Esquema de pines GPIO empleados en Hyot.



Anexo F

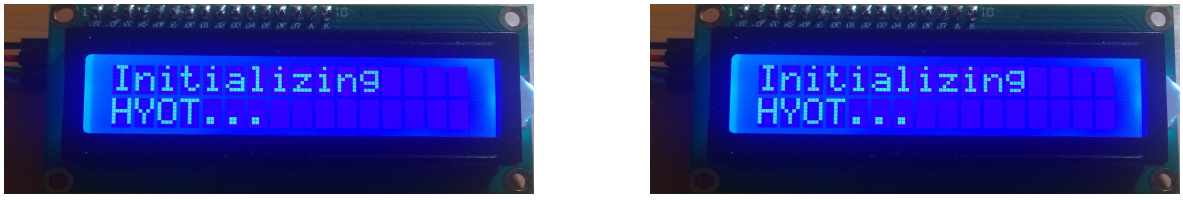
Información mostrada por el prototipo hardware de Hyot

El prototipo *hardware* de Hyot dispone de dos tipos de dispositivos de salida que muestran al usuario información en tiempo real sobre la ejecución del componente de monitorización de sucesos del entorno. Estos dos tipos de elementos son:

- Pantalla LCD (*Liquid Crystal Display*) 16x2. En concreto, se dispone de dos unidades donde cada una de ellas muestra la información procedente de un sensor siendo la configuración:
  - LCD 1 (pantalla izquierda) enlazada al sensor DHT-11.
  - LCD 2 (pantalla derecha) enlazada al sensor HC-SR04.
- LED (*Light-Emitting Diode*) de color rojo.

Cuando este componente se ejecuta, en primer lugar son efectuadas una serie de comprobaciones y configuraciones iniciales denotando tal estado por pantalla (Figura F.1).

Figura F.1: Prototipo hardware Hyot - Inicialización.



Una vez finalizada esta primera etapa, se inicia la monitorización de sucesos del entorno con la medición continua de los valores de los eventos de los sensores. Antes de efectuar la medición, se informa al usuario que va a comenzar tal tarea (Figura F.2) para posteriormente mostrar el sensor asociado a cada LCD (Figura F.3).

Figura F.2: Prototipo hardware Hyot - Comienzo de la medición.

Figura F.3: Prototipo hardware Hyot - Información del sensor.

En cada medición se toman tres valores: temperatura en grados centígrados (°C), humedad en porcentaje (rango entre 0-100 %) y distancia en centímetros (rango entre 2-150cm[1]). Las Figuras F.4 y F.5 muestran dos ejemplos de lectura de valores.

Figura F.4: Prototipo hardware Hyot - Ejemplo de medición.

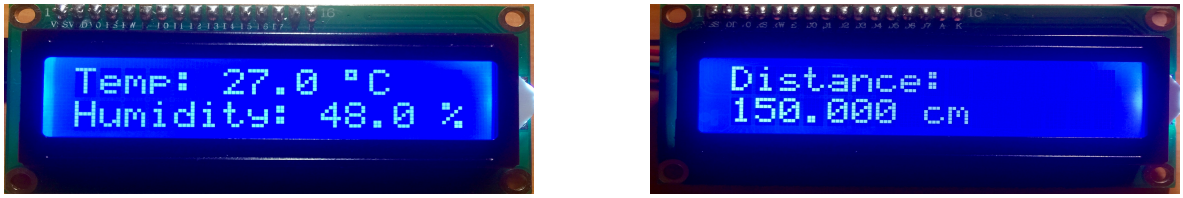
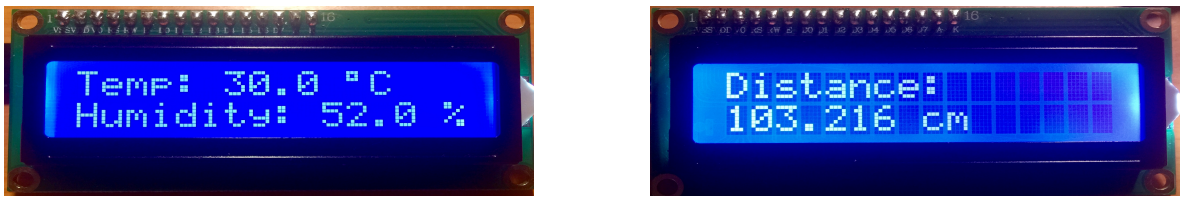


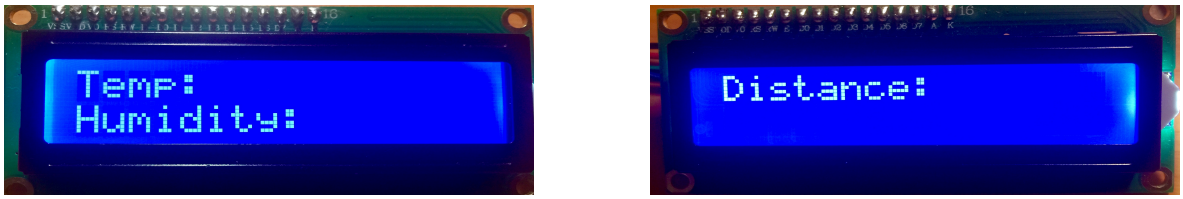
Figura F.5: Prototipo hardware Hyot - Otro ejemplo de medición.



En el caso de que se produzca un error de medición en alguno de los eventos por cualquier circunstancia y no se pueda obtener un valor, la pantalla se muestra vacía tal y como se presenta en la Figura F.6.

[1]El rango de distancia máxima de medición definida en la especificación del sensor es de 4 metros. Consultar [82] para obtener más información.

Figura F.6: Prototipo hardware Hyot - Ejemplo de medición errónea.



Cada medición realizada es analizada para determinar si presenta una lectura anómala y por tanto representa una incidencia en el entorno que hay que trazar mediante la activación del protocolo de alerta. En caso afirmativo, este protocolo se activa informando del evento que lo lanzó. La Figura F.7 muestra dos casos diferentes de activación de dicho procedimiento.

Figura F.7: Prototipo hardware Hyot - Protocolo de alerta activado por humedad y distancia.

Cuando este protocolo es activado se indica su inicio y su finalización al usuario (Figura F.8) y el LED es activado (Figura F.9) hasta que todos las acciones del protocolo son ejecutadas.

Figura F.8: Prototipo hardware Hyot - Inicio y finalización del protocolo de alerta.

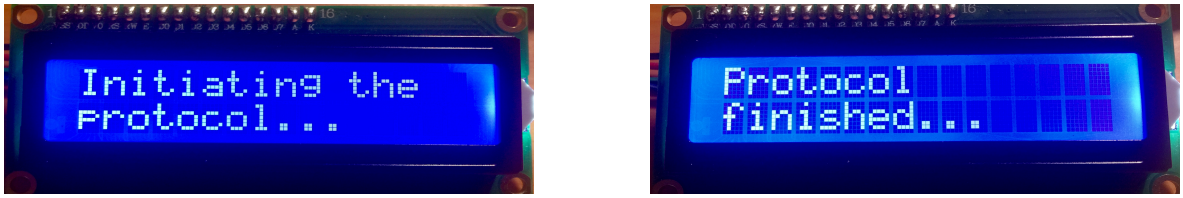
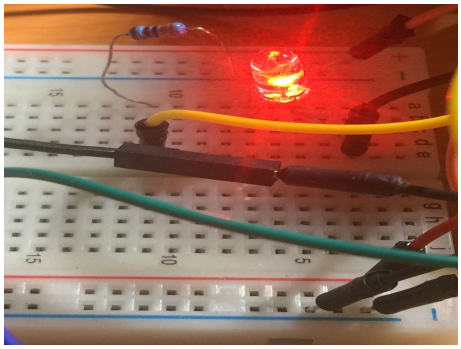


Figura F.9: Prototipo hardware Hyot - Activación del LED.



Anexo G

Manual de instalación y configuración

Todo proyecto requiere de un procedimiento de instalación y configuración previo para su puesta en marcha. Este manual expone el proceso para ejecutar Hyot así como los requerimientos necesarios para ello y las diferentes opciones de configuración.

G.1. Introducción

En este manual se documenta una guía que permite configurar y desplegar el proyecto Hyot en su totalidad, incluyendo los cinco componentes que lo conforman. Se debe tener en cuenta que el manual actual hace referencia a la configuración usada en el proyecto, siendo totalmente válido otras opciones que incluyan diferentes plataformas, servicios y/o cuentas de usuario aunque puede requerir la modificación de código.

Además, con el fin de evitar al lector del proceso de disposición de algunos de los requisitos generales, para la ejecución de prueba del proyecto se proporciona una configuración por defecto de los servicios a utilizar por lo que puede no ser necesario la creación de algunos requisitos mencionados a continuación.

G.2. Requisitos generales

La ejecución del proyecto en su totalidad conlleva la disposición de los siguientes entornos y servicios:

- Dispositivo Raspberry Pi[1] (RPi) junto con los diferentes componentes electrónicos que conforman el prototipo.

[1]Para el presente proyecto se ha utilizado la versión 3, siendo también válida cualquier versión anterior.

- Cuenta de usuario en el servicio IBM Cloud[2] [40] para desplegar los siguientes servicios de computación en la nube: Cloudant, Compose for MySQL y Liberty for Java.
- Cuenta de usuario en el servicio Microsoft Azure[3] para desplegar una máquina virtual (*Virtual Machine* -VM-).
- Cuenta de usuario en el servicio Gmail[4].
- Cuenta de usuario en el servicio Dropbox.

Además, es necesario la instalación de determinadas tecnologías. Sin embargo, se proporcionan mecanismos automatizados para ello o se utiliza un despliegue en la nube que omite parte de la instalación y configuración de éstas.

En el caso de desear configurar o modificar el componente que hace referencia al sistema web o incluso generar el artefacto WAR (*Web Application Archive*), se requiere tener instalado en el sistema local:

- Java JDK.
- *Framework* Grails.

G.3. Repositorio del proyecto Hyot

El código del proyecto Hyot se puede encontrar en el servicio GitHub:

- Clonar con el protocolo HTTPS (*Hypertext Transfer Protocol Secure*):
  `https://github.com/jesusiglesias/Hyot.git`
- Clonar con el protocolo SSH (*Secure Shell*):
  `git@github.com:jesusiglesias/Hyot.git`

G.4. Configuración e instalación

En esta sección se expone el procedimiento para configurar cada componente del proyecto y así poner en marcha Hyot.

[2]Si posee el rol de estudiante puede adquirir un código promocional de 6 meses en la dirección: `https://ibm.onthehub.com/`.

[3]Si posee el rol de estudiante puede crear una cuenta de este tipo en la dirección: `https://azure.microsoft.com/es-es/free/students/`.

[4]Para el funcionamiento de la funcionalidad de notificaciones es necesario permitir el acceso de aplicaciones menos seguras.

G.4.1. Componente - Configuración del dispositivo Raspberry Pi

En primer lugar, se debe configurar el dispositivo RPi para poder monitorizar los sucesos del entorno. Esta configuración se consigue ejecutando el componente de configuración del dispositivo RPi o en su defecto efectuar las acciones de forma manual.

Para iniciar este componente, desarrollado en el lenguaje Bash (*Bourne-again Shell*), basta con ejecutar el fichero `raspberrypi_setup.sh` -ubicado en el directorio `Hyot/hyot_raspberrypi/setup` según la jerarquía completa del proyecto- en la RPi desde un terminal con el comando `sudo bash`[5] y seguir los pasos indicados. Es de obligatoriedad que el dispositivo se encuentre conectado a la red para un funcionamiento correcto. Las acciones que efectúa este componente son:

1. Instalar y/o actualizar a la última versión los siguientes paquetes:

   - build-essential: contiene referencias a numerosos paquetes necesitados para el desarrollo y construcción de *software* en general.
   - gnupg: encriptación, desencriptación y firmado -entre otras funcionalidades- de cadenas de texto y/o ficheros usando GPG (*GNU Privacy Guard*).
   - i2c-tools: conjunto de controladores para la conexión de dispositivos mediante el protocolo I2C (*Inter-Integrated Circuit*).
   - python2.7: lenguaje de programación de uso general, orientado a objetos e interpretado.
   - python-dev: contiene todo lo necesario para compilar módulos y extensiones Python.
   - python-pip: sistema de gestión de paquetes utilizado para instalar y administrar paquetes de *software* escritos en Python.
   - python-smbus: enlace de Python para el acceso SMBus (*System Management Bus*) de Linux a través de `i2c-dev`. Este paquete es requerido para el uso de dispositivos conectados con el protocolo I2C.
   - rng-tools: conjunto de utilidades para la generación aleatoria de números con el objetivo de facilitar la entropia.

HYOT - PAQUETES	
Paquete	**Versión**
build-essential	12.3
gnupg	2.1.18-8
i2c-tools	3.1.2-3

[5] El término sudo permite delegar autoridad a usuarios normales para brindarles la capacidad de ejecutar algunas órdenes (o la totalidad) como usuario superprivilegiado.

python2.7	2.7.13-2
python-dev	2.7.13-2
python-pip	9.0.1-2
python-smbus	3.1.2-3
rng-tools	2-unofficial

Tabla G.1: Versiones empleadas de paquetes a fecha 13/08/2018

2. Instalar y/o actualizar a la última versión las siguientes librerías de Python:

   - Adafruit-DHT[6]: gestión del sensor DHT-11.
   - cloudant: interacción con el servicio de base de datos (BBDD) Cloudant NoSQL DB de IBM Cloud.
   - colorama: modificación del color y estilo del texto y fondo de la consola.
   - dropbox: interacción con el servicio Dropbox.
   - gpiozero: interfaz simple para dispositivos GPIO con RPi.
   - picamera: interfaz para el módulo de cámara de la RPi.
   - psutil: obtención de información sobre el sistema (CPU, memoria, discos, red, etc.) y los procesos en ejecución.
   - pysha3: implementación del algoritmo de funciones hash SHA3 (*Secure Hash Algorithm*) para Python.
   - python-gnupg: encriptación, desencriptación y firmado de cadenas de texto y/o ficheros usando GPG para Python.
   - PyYAML: parseador y analizador YAML (*YAML Ain't Markup Language*) para Python.
   - qrcode: generador de códigos QR.
   - RPi.GPIO: clase para controlar los pines de entrada/salida (*General Purpose Input Output* -GPIO-) en una RPi.
   - RPLCD: gestión de pantallas LCD (*Liquid Crystal Display*).
   - requests: implementación de peticiones HTTP (*Hypertext Transfer Protocol*) y HTTPS (*Hypertext Transfer Protocol Secure*).

[6]La instalación de esta librería al instante de desarrollar el código se realizó de forma manual descargando y descomprimiendo el paquete e instalándolo debido a que no existía una versión actualizada publicada en el repositorio PyPi. Por ese motivo se generó un *issue* en Github (`https://github.com/adafruit/Adafruit_Python_DHT/issues/77`). Posteriormente, este problema se solucionó y se modificó el código para realizar la instalación automática (`https://github.com/jesusiglesias/Hyot/commit/93353a618ec31187d75f0d7a0b0933d86e6c07be`).

HYOT - LIBRERÍAS PYTHON	
Librería	**Versión**
Adafruit-DHT	1.3.4
cloudant	2.8.1
colorama	0.3.9
dropbox	9.0.0
gpiozero	1.4.1
picamera	1.13
psutil	5.4.6
pysha3	1.0.2
python-gnupg	0.4.3
PyYAML	3.13
qrcode	6.0
requests	2.18.4
RPi.GPIO	0.6.3
RPLCD	1.1.0

Tabla G.2: Versiones empleadas de librerías Python a fecha 13/08/2018

3. Habilitar las siguientes interfaces:
    - i2c: bus de comunicación del protocolo I2C.
    - camera: bus de conexión eléctrico entre el procesador de la RPi y la cámara.
4. Preguntar al usuario si desea reiniciar el dispositivo como paso final lo cual es recomendable.

G.4.2. Componente - Protocolo de registro de incidencias

Este componente hace referencia al mecanismo de persistencia Blockchain (BC) implementado a través de la tecnología Hyperledger Fabric (HF) y accedido a través de la utilidad Hyperledger Composer (HC). Para realizar el despliegue tan solo basta con desplegar ambas utilidades en una máquina ya sea local o virtual, en un contenedor como, por ejemplo Kubernetes Service [39] o incluso un despliegue como servicio (*Blockchain as a Service* -BaaS- [41], [60]). En el caso de realizar el despliegue en una máquina local, el servidor de HC debería ser expuesto a la red para poder realizar las peticiones. Para evitar esto, se puede utilizar la herramienta Ngrok para tunelizar la comunicación hacia este servidor[7].

[7]El código del componente de monitorización de sucesos del entorno está preparado para soportar direcciones Ngrok.

El despliegue de ambas herramientas requiere de la instalación de una serie de prerequisitos y herramientas de línea de comandos que se pueden encontrar en la referencia [31]. Tras instalar las dependencias donde se incluye HF, el primer paso a realizar es iniciar esta BC para generar una *card*[8] con nombre `PeerAdmin@hlfv1.card` la cual posee privilegios para desplegar la red de negocio y es utilizada para este fin.

Posteriormente, utilizando esta *card* y la red de negocio generada a partir del código fuente con el comando: `npm build`[9] se procede a instalar y desplegar la red de negocio con los comandos:

```
composer network install --archiveFile [name]@[version].bna --card PeerAdmin@hlfv1

composer network start --networkName [name] --networkVersion [version] --card
   PeerAdmin@hlfv1 --networkAdmin admin --networkAdminEnrollSecret adminpw
```

Por último, únicamente faltaría arrancar el servidor de HC para que exponga el punto de acceso que permite interactuar con la BC. El comando utilizado configurando el uso de un certificado y *api-key* es:

```
composer-rest-server -c admin@hyot-network -n always -w true  -t -k [key] -e [
   certificate] -y [api-key]
```

G.4.3. Componente - Monitorización de sucesos del entorno

Este componente del proyecto Hyot contiene la funcionalidad núcleo puesto que permite la monitorización del entorno IoT y la actuación ante situaciones anómalas con el fin de generar una prueba o evidencia veraz e irrefutable del hecho. Para que este componente funcione correctamente es necesario:

- Haber ejecutado el componente de configuración del dispositivo RPi o en su defecto la ejecución manual de las acciones. Las librerías Python requeridas se encuentran definidas en un fichero de texto (`requirements.txt`) y pueden ser instaladas ejecutando el comando: `pip install -r requirements.txt` mientras que la habilitación de interfaces puede ser realizada con el comando `raspi-config`.
- Disponer del prototipo correctamente conectado.

[8] La *card* contiene el perfil de conexión el cual está compuesto de los detalles de identidad y los certificados y clave privada. Cada identidad es única y está asociada a un participante de la red existente, indicando a mayores la red a la que se puede conectar.

[9] La red de negocio resultante es un fichero `.bna` ubicado en el directorio `Hyot/hyot_bnd/dist` donde el nombre incluye la versión actual especificada.

- Obtener las direcciones I2C de los dos LCDs (*Liquid Crystal Display*) conectados a la RPi con el comando `i2cdetect` y en caso de diferir de las establecidas por defecto, deben ser indicadas en las opciones correspondientes a la hora de ejecutar el componente.
- Disponer de conexión a la red en la RPi.
- Proporcionar determinada información de configuración de servicios[10] y herramientas utilizadas mediante un fichero en formato YAML con nombre `hyot.yml` en un directorio llamado `conf` dentro del código de este componente, siendo la ruta completa `hyot_raspberrypi/hyot/conf` para la jerarquía completa del proyecto. Este fichero, que contiene la configuración de 5 utilidades empleadas, no está bajo el control de versiones por lo que el usuario debe crear una copia con la estructura indicada a continuación y ubicarlo en la ruta citada anteriormente. Ambos aspectos son comprobados durante la ejecución mostrando un error al usuario en caso de fallo.

```
| ESTRUCTURA DEL FICHERO DE CONFIGURACIÓN EN FORMATO YAML
| ------------------------------------------------------

# Utilidad GPG - Nombre e email para asociar al par de claves generado
gpg:
  name: [value]
  email: [value]
# Servicio Gmail - Credenciales de acceso de la dirección de correo origen
email:
  from: [value]
  password: [value]
# Servicio Cloudant NoSQL DB - Credenciales de acceso y URL
cloudant:
  username: [value]
  password: [value]
  url: [value]
# Servicio Drobpox - Token de acceso a la cuenta de Dropbox
dropbox:
  token: [value]
# Hyperledger Composer - Dirección IP y puerto donde está desplegado el servidor REST
  API[11] e indicación de si está securizado con un certificado autofirmado o con un
  certificado cuya identidad ha sido validada por una autoridad certificadora[12]
hl:
```

[10] En el caso de algunos servicios es necesario efectuar explícitamente una configuración desde la propia cuenta de usuario u obtener unas credenciales de acceso.

[11] En el caso de securizar este servidor con un *api-key* obtenido durante la inicialización de HC en este componente, este servidor debe ser arrancado posteriormente a la ejecución de este componente para poder indicar el valor. En caso contrario, el servidor HC puede ser arrancado con anterioridad.

[12] Esta propiedad es especificada para saltar el paso de verificación del certificado en las peticiones HTTPS ya

```
host: [value]
port: [value]
selfsignedcert: [True|False]
```

Para iniciar este componente, desarrollado en el lenguaje Python, basta con ejecutar el fichero `hyot_main.py` -ubicado en el directorio `Hyot/hyot_raspberrypi/hyot` según la jerarquía completa del proyecto- en la RPi desde un terminal con el comando `sudo python` y seguir los pasos indicados.

G.4.4. Componente - Desencriptación de evidencia

Este componente del proyecto Hyot permite efectuar la desencriptación, la verificación de firma y la comprobación de integridad de contenido de una evidencia previamente encriptada y firmada con la herramienta GPG por lo que su ejecución se produce en último lugar. Para poder ejecutar este componente desarrollado en lenguaje Python basta con ejecutar el fichero `hyot_decryption.py` -ubicado en el directorio `Hyot/hyot_raspberrypi/hyot_decryption` según la jerarquía completa del proyecto- en la RPi o en cualquier otro dispositivo electrónico –siempre y cuando disponga de las dependencias instaladas- desde un terminal con el comando `sudo python`. Las dependencias que requiere son:

- Haber ejecutado el componente de configuración del dispositivo RPi o en su defecto la ejecución manual del proceso de instalación de paquete y librerías Python. Éstas últimas se encuentran definidas en un fichero de texto (`requirements.txt`) y pueden ser instaladas ejecutando el comando: `pip install -r requirements.txt`.

- Disponer de la evidencia encriptada y firmada en el sistema local o en su defecto el enlace a Dropbox donde está almacenada.

- Disponer del par de claves utilizado durante la encriptación y firmado o en su defecto la huella digital (*fingerprint*) asociado al par de claves empleado.

- Disponer del código *hash* del contenido de la evidencia original, obtenible del mecanismo de persistencia de BC.

- Disponer de conexión a la red en la RPi en caso de querer descargar la evidencia desde el servicio de almacenamiento en la nube.

que si se usa un certificado autofirmado al no ser capaz de verificar la identidad, la petición es denegada. En caso de usar un certificado validado por una CA, la verificación se habilita. Consultar [70] para obtener más información.

G.4.5. Componente - Sistema web

La puesta en marcha del sistema web únicamente requiere, a partir del código fuente, de la generación del artefacto WAR el cual debe ser desplegado posteriormente en un contenedor de aplicaciones como es Tomcat o JBoss, por ejemplo. Para ello, basta con ejecutar el comando de Grails: `grails war`. El artefacto generado por defecto lo hace con la configuración para el entorno de producción y se ubica en la ruta: `hyot_app/build/libs`.

Este sistema web puede requerir la modificación de determinadas configuraciones para su adaptación. En el fichero `hyot_app/grails-app/conf/application.yml` se puede modificar, entre otros detalles:

- Cuenta Gmail y credenciales.
- Dirección donde el servidor de Hyperledger Composer expone la API.
- Dirección y credenciales del servicio Cloudant NoSQL DB.
- Dirección y credenciales de la BBDD del sistema web.

G.5. Generación del par de claves y certificado SSL autofirmado

La securización de la comunicación entre diferentes partes donde se desconoce la confiabilidad de cada una de ellas y del medio de comunicación, provoca que sea indispensable la securización de ésta para que la información transmitida se encuentre en todo momento encriptada y por tanto protegida. En el proyecto Hyot, se ha habilitado HTTPS y TLS (*Transport Layer Security*) tanto en el servidor de HC como en el sistema web. Para ello, es necesario generar en primer lugar un par de claves (clave pública y privada) y un certificado autofirmado, por ejemplo desde una máquina local.

Este procedimiento, descrito a continuación, se ha llevado a cabo con la herramienta OpenSSL la cual es un proyecto de *software* libre que contiene un conjunto robusto de herramientas de administración y bibliotecas de propósito general relacionadas con la criptografía para los protocolos TLS y SSL (*Secure Sockets Layer*).

1. **Generación de la clave pública y privada (Figura G.1):** utilizando el algoritmo de criptografía asimétrica RSA (*Rivest, Shamir y Adleman*) con una longitud de 2048 bits[13] y

[13]Los estándares de seguridad indican que los certificados SSL deben ser generados con un tamaño de clave de al menos 2048 bits para aumentar la dificultad de ser vulnerables ante un posible ataque.

encriptada con el estándar de criptografía simétrica AES (*Advanced Encryption Standard*) con longitud de clave 256 bits. Durante la generación se solicita una contraseña[14] de protección para la clave privada. El fichero resultante: `hyot.pem` contiene el par de claves, es decir, tanto la clave privada como la clave pública, siendo la primera de ellas la base de toda confianza por lo que es muy importante su protección y no revelación a personas ajenas.

Figura G.1: Certificado SSL - Generación del par de claves.

```
MacBook-Pro-de-Jesus:cert jesusiglesias$ openssl genrsa -aes256 -out hyot.pem 2048
Generating RSA private key, 2048 bit long modulus
..............+++
....+++
e is 65537 (0x10001)
Enter pass phrase for hyot.pem:
Verifying - Enter pass phrase for hyot.pem:
```

2. **Extracción de la clave pública[15] (Figura G.2):** al fichero `hyot_public.pem`. Esta clave es aquella que se puede distribuir libremente ya sea para realizar el proceso de encriptación o firma digital.

Figura G.2: Certificado SSL - Extracción de la clave pública.

```
MacBook-Pro-de-Jesus:cert jesusiglesias$ openssl rsa -in hyot.pem -outform PEM -pubout -out hyot_public.pem
Enter pass phrase for hyot.pem:
writing RSA key
```

3. **Obtención de la clave privada sin protección de contraseña (Figura G.3):** este paso solamente es necesario para securizar el servidor de HC debido a que actualmente (v0.19.10 a fecha 22/06/2018) no acepta el uso de una clave privada protegida con contraseña[16]. Sin embargo, en cualquier otra situación (p. ej. el servidor del cliente web) se aconseja introducir una contraseña para proteger dicha clave. El fichero `hyot_nopassphrase.pem` contiene la clave privada pero sin protección de contraseña.

[14] Se aconseja la utilización de contraseñas robustas.

[15] Este paso no es necesario realizarlo aunque se indica para el conocimiento del lector.

[16] El siguiente *issue* en GitHub hace referencia a esta cuestión. Link: `https://github.com/hyperledger/composer/issues/3132`

Figura G.3: Certificado SSL - Obtención de clave privada sin protección de contraseña.

```
MacBook-Pro-de-Jesus:cert jesusiglesias$ openssl rsa -in hyot.pem -out hyot_nopassphrase.pem
Enter pass phrase for hyot.pem:
writing RSA key
```

4. **Generación de la solicitud de firma de certificado (*Certificate Signing Request* -CSR-) (Figura G.4):** a partir de la clave privada. Se utiliza como algoritmo de firma la función hash recomendada (SHA-256) y no el de por defecto (SHA-1) ya que se detectaron colisiones y se considera inseguro [57].

Figura G.4: Certificado SSL - Solicitud de firma de certificado.

```
MacBook-Pro-de-Jesus:cert jesusiglesias$ openssl req -new -key hyot.pem -out hyot_cert.csr -sha256 -config ssl.cnf
Enter pass phrase for hyot.pem:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) []:ES
State or Province Name (full name) []:Madrid
Locality Name (eg, city) []:Madrid
Organization Name (eg, company) []:Hyot
Organizational Unit Name (eg, section) []:
Common Name (eg, fully qualified host name) []:40.121.15.202
Email Address []:hyot.project@gmail.com
```

Durante la generación se solicitan una serie de datos[17] los cuales deben ser introducidos correctamente con especial énfasis el campo *Common Name* que hace referencia al dominio o dirección IP del servidor para que a la hora de solicitar una petición HTTPS al servidor el cual debe presentar el certificado SSL -a partir de ahora como certificado-, se evite la advertencia *NET::ERR_CERT_COMMON_NAME_INVALID*.

Recientemente, para los usuarios del navegador Google Chrome con una versión posterior a la 58, liberada en marzo de 2017, la advertencia anteriormente mencionada comenzó a aparecer en aquellos certificados válidos donde solamente se especificaba el campo *Common*

[17]El ejemplo expuesto hace referencia a la generación de solicitud para el servidor de HC desplegado en una máquina virtual (*Virtual Machine* -VM-) en el servicio Microsoft Azure por lo que el dato *Common Name* debe adaptarse para otros casos como, por ejemplo el cliente web donde el dominio o dirección IP será diferente.

Name, el cual empezó a ser ignorado [78]. Hasta dicha versión, el certificado con este campo correctamente definido no presentaba ningún problema. Para evitar esta situación, basta con definir valores en el campo *Subject Alternative Name* en un fichero de configuración y cargarlo a la hora de generar la solicitud de firma y del posterior proceso de autofirmado [13]. A continuación, se detalla la estructura del fichero de configuración `ssl.cnf` donde lo único que hay que introducir es el valor del campo `alt_names` con la dirección IP del servidor.

```
| ESTRUCTURA DEL FICHERO DE CONFIGURACIÓN SSL - SSL.CNF
| ----------------------------------------------------

[ req ]
distinguished_name          = req_distinguished_name
req_extensions              = req_ext

[ req_distinguished_name ]
countryName                 = Country Name (2 letter code)
stateOrProvinceName         = State or Province Name (full name)
localityName                = Locality Name (eg, city)
organizationName            = Organization Name (eg, company)
organizationalUnitName      = Organizational Unit Name (eg, section)
commonName                  = Common Name (eg, fully qualified host name)
emailAddress                = Email Address

[ req_ext ]
subjectAltName              = @alt_names

[ alt_names ]
IP.1                        = [IP Server]

-----------------------------------------------------
# INFORMACIÓN ADICIONAL

# Si en lugar de una dirección IP, fuese un dominio
[ alt_names ]
DNS.1                       = [Domain]

-----------------------------------------------------
# También, se pueden especificar varias alternativas
[ alt_names ]
DNS.1                       = [Domain1]
DNS.2                       = [Domain2]
IP.1                        = [IP Server]
```

El fichero resultante de la solicitud de firma `hyot_cert.csr` debería ser enviado a una CA la

cual verificaría la identidad del solicitante y expediría un certificado firmado. En este caso, como se va a utilizar un certificado autofirmado no es necesario este proceso.

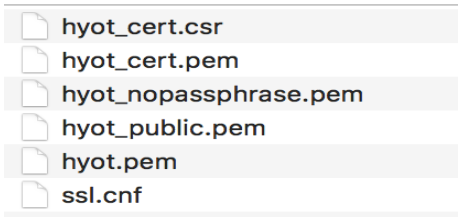
5. **Generación del certificado autofirmado[18] (Figura G.5):** con una validez de 1 año. Además, como se mencionó en el paso anterior a la hora de autofirmar el certificado también es necesario cargar la configuración del fichero `ssl.cnf`.

Figura G.5: Certificado SSL - Proceso de autofirmado.

```
MacBook-Pro-de-Jesus:cert jesusiglesias$ openssl x509 -sha256 -req -days 365 -in hyot_cert.csr -signkey hyot.pem
-out hyot_cert.pem -extensions req_ext -extfile ssl.cnf
Signature ok
subject=/C=ES/ST=Madrid/L=Madrid/O=Hyot/CN=40.121.15.202/emailAddress=hyot.project@gmail.com
Getting Private key
Enter pass phrase for hyot.pem:
```

A modo de síntesis, en la Figura G.6 se muestran los ficheros resultantes de este proceso: 5 ficheros relacionados con el proceso de generación del certificado y un fichero de configuración.

Figura G.6: Ficheros resultantes del proceso de generación del certificado autofirmado.



G.6. Verificar conexión HTTPS

Una vez desplegado un servidor con el protocolo HTTPS se puede verificar de una forma sencilla que la comunicación es segura[19] [20]. La primera vez que se accede al sitio web del servidor, el navega-

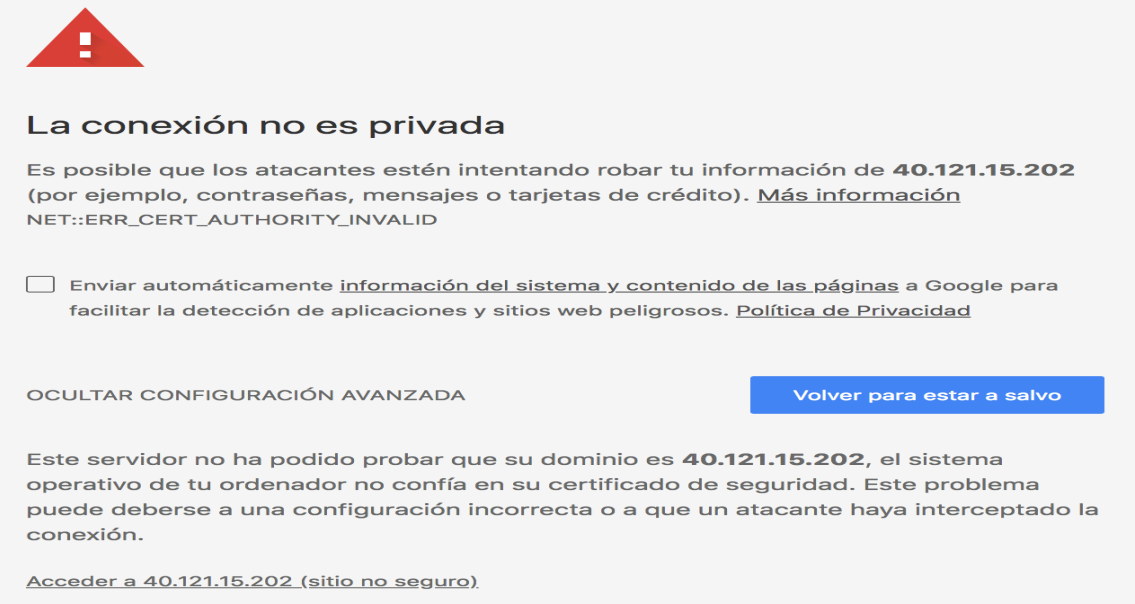
[18] Para generar un certificado autofirmado no es necesario generar la solicitud de firma ya que puede generarse directamente a partir de la clave privada. Sin embargo, con el método expuesto se evita realizar todo el procedimiento de nuevo en el caso de necesitar un certificado firmado por una CA ya que la solicitud de firma ya fue generada y puede ser enviada directamente a la entidad certificadora. Si aun así se desea generar directamente, se puede utilizar el comando: `openssl req -new -x509 -days 365 -key [hyot.pem] -out [hyot_cert.pem] -config ssl.cnf`

[19] Los pasos indicados hacen referencia al procedimiento seguido para el navegador Google Chrome en un SO MacOS. Otros navegadores y/o plataformas pueden diferir en el procedimiento.

[20] El procedimiento a continuación descrito también es aplicable a cualquier servidor en general como el presente en el sistema web de Hyot.

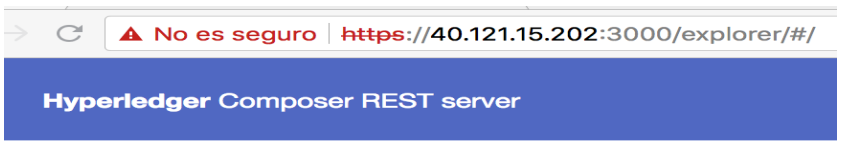
dor muestra una alerta (Figura G.7) indicando la advertencia *NET::ERR_CERT_AUTHORITY_INVALID* la cual significa que el certificado no es de confianza ya que no se puede encontrar una identificación firmada por una CA.

Figura G.7: Advertencia de conexión HTTPS no confiable.



Aceptando acceder aun con el mensaje de advertencia se puede observar como en la barra de direcciones aparece el símbolo HTTPS tachado y en rojo (Figura G.8), signo de lo indicado anteriormente. Sin embargo, esto no quiere decir que la comunicación se encuentre sobre una conexión insegura.

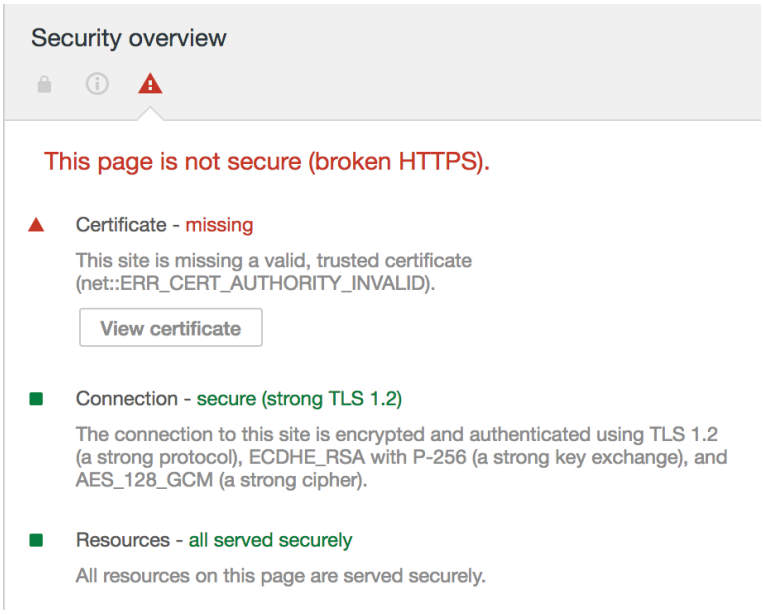
Figura G.8: Conexión HTTPS no confiable.



La navegación en este momento es totalmente segura y la información se transmite de forma encriptada, como se muestra en la Figura G.9[21] donde se indica que todos los recursos son servidos de manera segura y se utiliza el protocolo TLS, entre otros detalles.

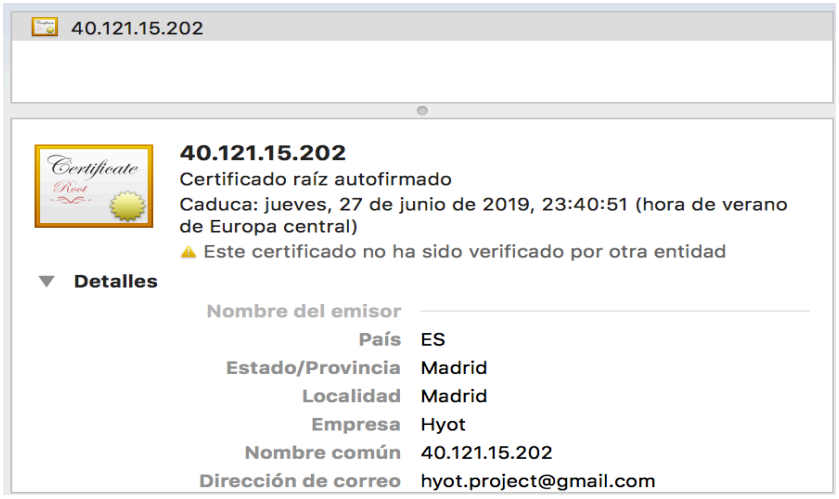
[21]Captura de pantalla obtenida desde el panel *Security* de las Herramientas para desarrolladores de Google Chrome.

Figura G.9: Vista general de seguridad antes de confiar en el certificado.



Para hacer más elegante la navegación del usuario mostrando el símbolo HTTPS en verde (signo de una total confianza), lo que se debe realizar es importar el certificado autofirmado en el almacén de certificados del ordenador desde el cual se está navegando[22] (Figura G.10). Este paso se omite cuando el certificado se encuentra firmado por una CA ya que los ordenadores y navegadores traen preinstalados por defecto almacenes de certificados de diversas empresas.

Figura G.10: Instalación del certificado autofirmado en el almacén de certificados del ordenador.



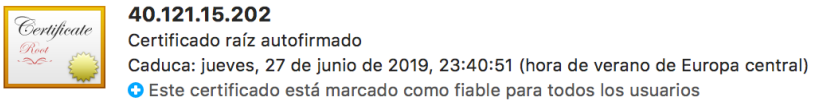
[22]Esta acción únicamente puede ser efectuada por usuarios que posean el certificado.

Como se observa de la figura anterior, una vez que se instaló el certificado el sistema avisa que no es confiable ya que no fue verificado por una CA. Para marcarlo como fiable para el usuario, se debe indicar explícitamente (Figura G.11).

Figura G.11: Confiando en el certificado autofirmado.

Con el certificado marcado como confiable ya no aparece ninguna advertencia (Figura G.12).

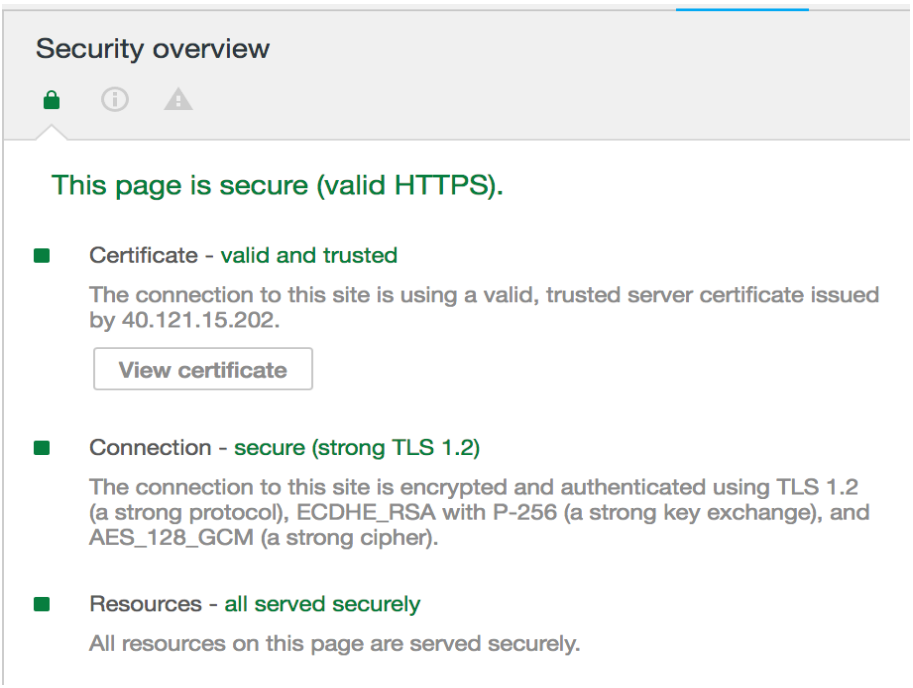
Figura G.12: Certificado autofirmado confiable.



Si se accede de nuevo al sitio web se puede observa como ya aparece el símbolo HTTPS en verde (Figura G.13) y ha desaparecido la advertencia *NET::ERR_CERT_AUTHORITY_INVALID* (Figura G.14).

Figura G.13: Conexión HTTPS confiable.

Figura G.14: Vista general de seguridad antes de confiar en el certificado.



G.7. Credenciales de acceso

A continuación se indica la credencial de acceso del usuario administrador creado por defecto para el sistema web:

- Usuario: hyot_admin / hyot.project@gmail.com
- Contraseña: Qwerty321!

G.8. Ayuda

Si desconoce cómo usar o configurar las herramientas anteriormente mencionadas o necesita información adicional, se recomienda consultar los siguientes recursos para la resolución de dudas y preguntas:

- **Documentación de Grails:**

  `http://grails.org/documentation.html`

- **Documentación de Hyperledger Composer:**

  `https://hyperledger.github.io/composer/latest/`

- **Documentación de Hyperledger Fabric:**

  `https://hyperledger-fabric.readthedocs.io/en/release-1.2/`

- **Documentación de IBM Cloud:**

  `https://console.bluemix.net/docs/`

- **Documentación de Microsoft Azure:**

  `https://docs.microsoft.com/es-es/azure/`

- **Documentación de Python 2.7:**

  `https://docs.python.org/2/index.html`

- **Documentación de Raspberry Pi:**

  `https://www.raspberrypi.org/documentation/`

G.9. Contacto y versión

Para cuestiones directamente relacionadas al proyecto **Hyot**, puede ponerse en contacto con el autor vía *email* en la siguiente dirección: `jesusgiglesias@gmail.com`.

La versión actual del presente documento a fecha de 03 de septiembre de 2018: **1.0.0**.

Anexo H

Manual de usuario

Hyot es una prueba de concepto (*Proof of Concept* -PoC-) de código abierto para la trazabilidad de un entorno controlado del Internet de las Cosas (*Internet of Things* -IoT-) mediante la tecnología Hyperledger Fabric (HF). El objetivo de esta solución es la monitorización de dicho entorno para el reconocimiento de sucesos anómalos en cuyo caso el propósito es garantizar una prueba irrefutable y veraz de cada incidencia producida. Esto es posible gracias a la constante recolección de sucesos por medio de sensores situados en una Raspberry Pi (RPi) 3 y la inclusión de incidencias en la Blockchain (BC) de HF aprovechando de tal forma las ventajas que brinda esta tecnología. Asimismo, mediante un sistema web se puede consultar la información recolectada en tiempo real.

Cinco son los componentes que conforman Hyot, y a continuación se detalla el modo de funcionamiento de cada uno -excepto el componente de protocolo de registro de incidencias el cual ya se encuentra embebido en otros componentes- y la variedad de funcionalidades que ofrece al usuario.

H.1. Componente de configuración del dispositivo Raspberry Pi

Este componente del proyecto Hyot permite efectuar la configuración inicial del dispositivo RPi para posteriormente ejecutar correctamente el componente de monitorización de sucesos del entorno. Su uso es totalmente opcional puesto que los pasos que son efectuados pueden ser ejecutados de forma manual. Sin embargo, con el fin de facilitar la puesta en marcha de Hyot este proceso se automatiza proporcionando un *script* desarrollado en el lenguaje Bash (*Bourne-again Shell*).

Para iniciar este componente basta con ejecutar el fichero `raspberrypi_setup.sh` -ubicado

en el directorio `Hyot/hyot_raspberrypi/setup` según la jerarquía completa del proyecto- en la RPi desde un terminal con el comando `sudo bash`[1] y seguir los pasos indicados. La configuración, que requiere de conexión a la red, se compone de tres fases ejecutadas en el siguiente orden:

1. Comprobaciones iniciales.
2. Instalación de dependencias requeridas (paquetes y librerías Python).
3. Activación de interfaces.

Además, este componente ofrece un menú de ayuda, opción `-h` o `--help`, donde se detallan todas las opciones disponibles (Figura H.1), siendo las definidas las mostradas a continuación:

- Opción `-v` o `--verbose`: muestra más detalle sobre las operaciones que se van ejecutando.
- Opción `-p` o `--packages`: ejecuta solamente la fase de instalación de paquetes y librerías.
- Opción `-i` o `--interfaces`: ejecuta solamente la fase de activación de interfaces.

Figura H.1: Configuración inicial de la RPi - Menú de ayuda.

```
pi@raspberrypi:~/Desktop/hyot_raspberrypi/setup $ sudo bash raspberrypi_setup.sh -h

 _    ___     ______ _______
| |  | \ \   / / __ \__   __|
| |__| |\ \_/ / |  | | | |
|  __  | \   /| |  | | | |
| |  | |  | | | |__| | | |
|_|  |_|  |_|  \____/  |_|

A PoC for traceability in IoT environments through Hyperledger Fabric by:

    - Jesús Iglesias García, jesusgiglesias@gmail.com

-------------------------------------------------------

HYOT - RASPBERRY PI SETUP

This component sets up the initial configuration on the Raspberry Pi to run the component
monitoring of environmental events.

Type the '-h' or '--help' option to get more information.


====================  HELP  ====================

USAGE:    sudo raspberrypi_setup.sh [--help] [[--interfaces|--packages][--verbose]]

BASIC OPTIONS:

   -h, --help                Shows the help.
   -i, --interfaces          Runs only the activation of interfaces.
   -p, --packages            Runs only the installation of packages and libraries.
   -v, --verbose             Provides very helpful additional details as to what the code is doing.
```

[1] El término `sudo` permite delegar autoridad a usuarios normales para brindarles la capacidad de ejecutar algunas órdenes (o la totalidad) como usuario superprivilegiado.

H.1.1. Comprobaciones iniciales

En esta primera fase se realizan una serie de comprobaciones para garantizar que la posterior ejecución se efectúe de manera correcta y sin ningún contratiempo. Si alguna de las comprobaciones falla ya sea porque no cumple la condición, porque el comando a usar no se encuentra instalado en la RPi o por cualquier otra causa, se muestra por consola un error al usuario y se finaliza la ejecución. Las comprobaciones incluyen:

1. Carga del fichero de utilidades `utils.sh`, ubicado en el mismo directorio (Figura H.2).

Figura H.2: Configuración inicial de la RPi - Carga de utilidades.

```
pi@raspberrypi:~/Desktop/hyot_raspberrypi/setup $ bash raspberrypi_setup.sh
raspberrypi_setup.sh: line 71: /home/pi/Desktop/hyot_raspberrypi/setup/utils.sh: No such file or directory
```

2. Verificación de ejecución con un usuario superprivilegiado (Figura H.3).

Figura H.3: Configuración inicial de la RPi - Verificación de usuario.

```
pi@raspberrypi:~/Desktop/hyot_raspberrypi/setup $ bash raspberrypi_setup.sh
✖ This component must be run as root.
```

3. Verificación de ejecución en una plataforma GNU/Linux. La Figura H.4 muestra la ejecución del código en un sistema con MacOS (*Macintosh Operating System*).

Figura H.4: Configuración inicial de la RPi - Verificación de plataforma.

```
MacBook-Pro-de-Jesus:setup jesusiglesias$ sudo bash raspberrypi_setup.sh
✖ This component must be run on GNU/Linux platform (e.g. Raspbian).
```

4. Verificación de ejecución en un dispositivo RPi, en cualquiera de sus versiones. La Figura H.5 muestra el error cuando el código se ejecuta en un sistema operativo Debian, el cual se basa en la plataforma GNU/Linux pero no es una RPi.

Figura H.5: Configuración inicial de la RPi - Verificación de dispositivo.

```
jesusiglesias_debian@debian-gnu-linux-vm:~/Desktop/hyot_raspberrypi/setup$ sudo bash raspberrypi_setup.sh
✖ This component must be run on a Raspberry Pi.
```

5. Verificación de conexión a Internet mediante la ejecución de un comando *ping* al buscador de Google. La Figura H.6 muestra un ejemplo de comprobación de existencia de un comando en el sistema local y la Figura H.7 la ejecución del comando.

Figura H.6: Configuración inicial de la RPi - Verificación de comando.

```
pi@raspberrypi:~/Desktop/hyot_raspberrypi/setup $ sudo bash raspberrypi_setup.sh
✖ Command not found: wget. Please, install this command to check if the network connection is available.
```

Figura H.7: Configuración inicial de la RPi - Verificación de conexión a Internet.

```
pi@raspberrypi:~/Desktop/hyot_raspberrypi/setup $ sudo bash raspberrypi_setup.sh
✖ Raspberry Pi is not connected to the network. Please, enable the network to continue the setup.
```

6. Verificación de que no existe otra instancia de este componente en ejecución (Figura H.8). En este paso también se comprueba si el comando `pgrep` se encuentra instalado -caso similar al mostrado en el paso anterior-.

Figura H.8: Configuración inicial de la RPi - Verificación de concurrencia.

```
pi@raspberrypi:~/Desktop/hyot_raspberrypi/setup $ sudo bash raspberrypi_setup.sh
✖ Process: raspberrypi_setup.sh is already running with PID 5931.
```

7. Comprobación de las opciones introducidas, donde se valida:

   - El número de opciones, debiendo introducir una, dos o ninguna opción (Figura H.9).
   - La validez de cada opción indicada (Figura H.10).

Figura H.9: Configuración inicial de la RPi - Verificación del número de opciones.

```
[pi@raspberrypi:~/Desktop/hyot_raspberrypi/setup $ sudo bash raspberrypi_setup.sh -v -i -p

 _    ___     ______ _______
| |  | \ \   / / __ \__   __|
| |__| |\ \_/ / |  | | | |
|  __  | \   /| |  | | | |
| |  | |  | | | |__| | | |
|_|  |_|  |_|  \____/  |_|

A PoC for traceability in IoT environments through Hyperledger Fabric by:

    - Jesús Iglesias García, jesusgiglesias@gmail.com

-------------------------------------------------------

HYOT - RASPBERRY PI SETUP

This component sets up the initial configuration on the Raspberry Pi to run the component
monitoring of environmental events.

Type the '-h' or '--help' option to get more information.

✖ Invalid parameter number. Please, type the option '-h' or '--help' to show the help.
```

Figura H.10: Configuración inicial de la RPi - Verificación de opciones.

```
[pi@raspberrypi:~/Desktop/hyot_raspberrypi/setup $ sudo bash raspberrypi_setup.sh -a

 _    ___     ______ _______
| |  | \ \   / / __ \__   __|
| |__| |\ \_/ / |  | | | |
|  __  | \   /| |  | | | |
| |  | |  | | | |__| | | |
|_|  |_|  |_|  \____/  |_|

A PoC for traceability in IoT environments through Hyperledger Fabric by:

    - Jesús Iglesias García, jesusgiglesias@gmail.com

-------------------------------------------------------

HYOT - RASPBERRY PI SETUP

This component sets up the initial configuration on the Raspberry Pi to run the component
monitoring of environmental events.

Type the '-h' or '--help' option to get more information.

✖ Unknown option: -a. Please, type the option '-h' or '--help' to show the help.
```

Por último, mencionar que la señal de terminación (Ctrl + C) del proceso cuando está en ejecución también es capturada, informando al usuario de que debería esperar hasta que la ejecución finalizase de forma autónoma (Figura H.11).

Figura H.11: Configuración inicial de la RPi - Señal de terminación de la ejecución.

```
pi@raspberrypi:~/Desktop/hyot_raspberrypi/setup $ sudo bash raspberrypi_setup.sh
^C
✖ Exception: KeyboardInterrupt. Please, wait until the process finishes.
```

H.1.2. Instalación de dependencias requeridas

Esta segunda fase se encarga de instalar y/o actualizar los paquetes del sistema y las librerías Python empleadas por el resto de componentes ejecutados en la RPi. Ambas secciones ejecutan las mismas acciones con la salvedad de adecuar los comandos empleados para cada una. En primer lugar, se comprueba si el paquete o la librería se encuentra ya instalada en el sistema, dando lugar a la siguiente bifurcación:

- En caso afirmativo, se comprueba si está actualizado/a no realizando ninguna acción en dicho caso (Figuras H.12 y H.13), y ejecutando el proceso de actualización en caso contrario (Figura H.14). La Figura H.15 muestra la notificación al usuario de un error producido durante el procedimiento de actualización.

Figura H.12: Configuración inicial de la RPi - Paquetes ya instalados y actualizados.

```
Starting the configuration...

Packages
------------------------------------------------------------------

➜ Package: python2.7 is installed and updated in the system.
➜ Package: build-essential is installed and updated in the system.
```

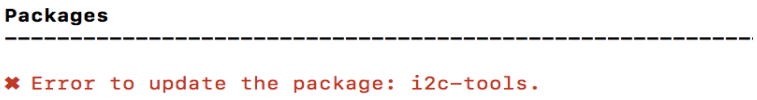
Figura H.13: Configuración inicial de la RPi - Librerías ya instaladas y actualizadas - Modo verbose.

```
Python libraries
------------------------------------------------------------------

Checking if the 'Adafruit-DHT' library is installed and updated.
Library is installed. Checking if this one is updated.
Library is already updated to the last version.
➜ Library: Adafruit-DHT is installed and updated in the system.
```

Figura H.14: Configuración inicial de la RPi - Actualizar librerías - Modo verbose.

```
Python libraries
------------------------------------------------------------------

Checking if the 'Adafruit-DHT' library is installed and updated.
Library is installed. Checking if this one is updated.
Library should be updated. Updating...
➜ Library: Adafruit-DHT is installed and updated in the system.
```

Figura H.15: Configuración inicial de la RPi - Error durante la actualización.



- En otro caso, se comprueba si existe en el repositorio de tal forma que realiza la instalación en caso afirmativo (Figuras H.16 y H.17) o no realiza ninguna acción más que notificar al usuario en caso de que no exista (Figura H.18). La Figura H.19 muestra la notificación mostrada cuando ocurre un error durante la instalación.

Figura H.16: Configuración inicial de la RPi - Instalar paquetes.

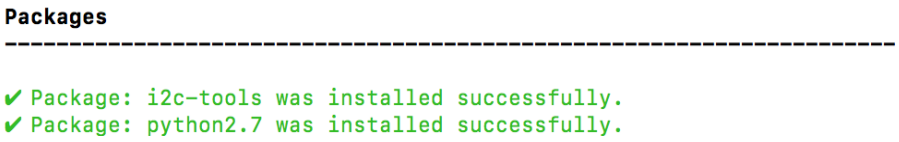


Figura H.17: Configuración inicial de la RPi - Instalar librerías - Modo verbose.

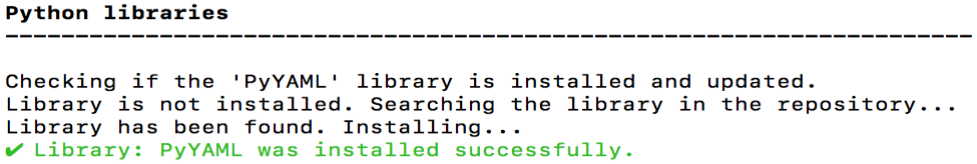


Figura H.18: Configuración inicial de la RPi - Librería no instalada e inexistente.

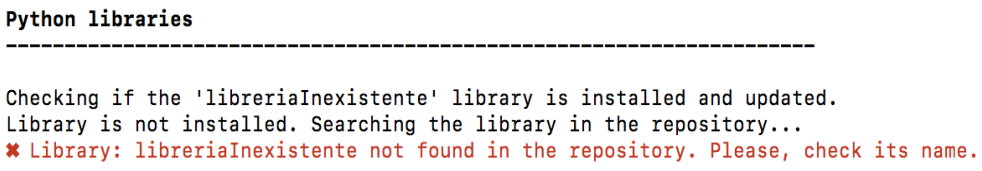
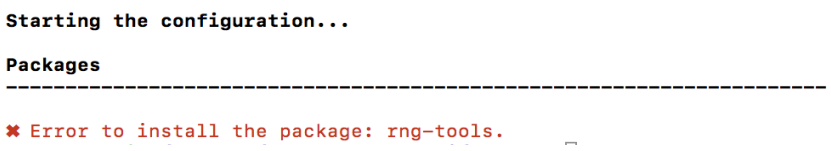


Figura H.19: Configuración inicial de la RPi - Error durante la instalación.



H.1.3. Activación de interfaces

Las interfaces `Camera` e `I2C` de la RPi es necesario que se encuentren habilitadas a la hora de ejecutar el componente de monitorización de sucesos del entorno para permitir que estas dos funcionalidades funcionen de forma correcta. En esta última fase se persigue esta finalidad donde en primer lugar se comprueba si la interfaz ya se encuentra habilitada. En caso afirmativo, ninguna acción es realizada (Figuras H.20 y H.21).

Figura H.20: Configuración inicial de la RPi - Interfaces habilitadas.

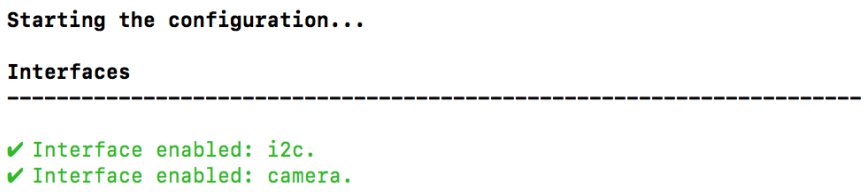
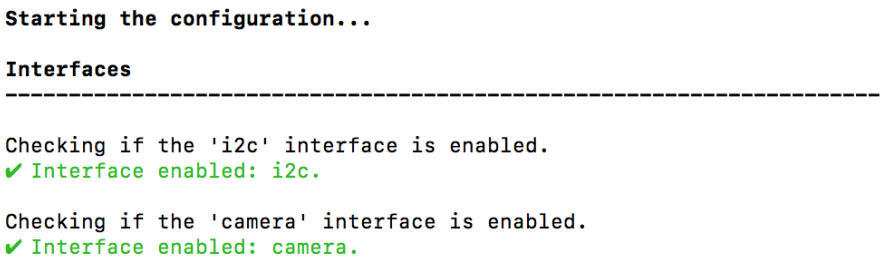


Figura H.21: Configuración inicial de la RPi - Interfaces habilitadas - Modo verbose.



En caso contrario, es decir, si la interfaz se encuentra deshabilitada se procede a su activación (Figura H.22).

Figura H.22: Configuración inicial de la RPi - Habilitar interfaces - Modo verbose.

```
Starting the configuration...

Interfaces
------------------------------------------------------------------------

Checking if the 'i2c' interface is enabled.
Interface disabled. Enabling...
✔ Interface enabled: i2c.

Checking if the 'camera' interface is enabled.
Interface disabled. Enabling...
✔ Interface enabled: camera.
```

Si durante la activación se produjese algún error en alguna de las interfaces, se notifica al usuario (Figuras H.23 y H.24).

Figura H.23: Configuración inicial de la RPi - Error durante la activación de interfaces.

```
Starting the configuration...

Interfaces
------------------------------------------------------------------------

✖ Error to enable the interface: i2c.
```

Figura H.24: Configuración inicial de la RPi - Error durante la activación de interfaces - Modo verbose.

```
Starting the configuration...

Interfaces
------------------------------------------------------------------------

Checking if the 'i2c' interface is enabled.
Interface disabled. Enabling...
✖ Error to enable the interface: i2c.
```

La comprobación, activación y desactivación de las interfaces se realiza con el comando `raspi-config` por lo que si este comando por cualquier causa no se encontrara instalado en el dispositivo se muestra el siguiente error (Figura H.25).

Figura H.25: Configuración inicial de la RPi - Comando raspi-config no instalado.

Por último, se informa al usuario de la finalización del proceso y de los siguientes pasos a realizar para la continuación de ejecución de Hyot junto con la cuestión al usuario de si desea reiniciar el dispositivo lo cual es recomendable. En caso afirmativo, el dispositivo se reinicia tras 5 segundos (Figura H.26) por lo que previamente se comprueba que el comando para realizar esta acción se encuentre instalado en el sistema ya que en caso contrario se produciría el error mostrado en la Figura H.27.

Figura H.26: Configuración inicial de la RPi - Reiniciar dispositivo.

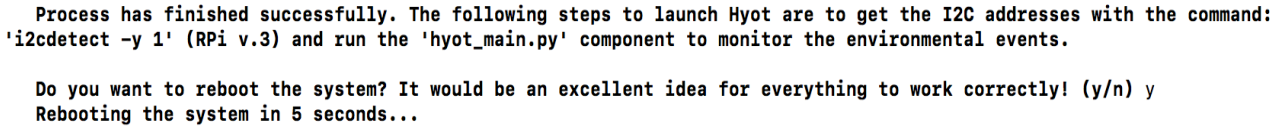
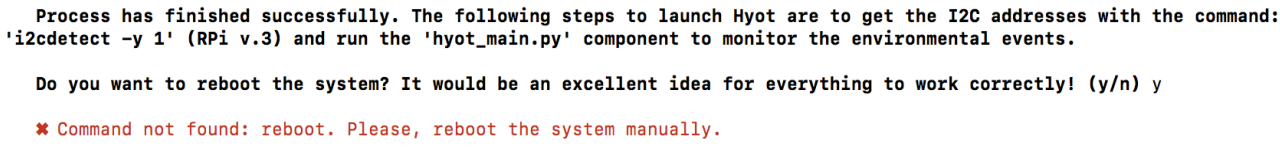


Figura H.27: Configuración inicial de la RPi - Comando reboot.



Uno de los pasos indicados a realizar a continuación es obtener las direcciones I2C (*Inter-Integrated Circuit*) de los dos LCDs (*Liquid Crystal Display*) conectados a la RPi las cuales deben ser indicadas en el siguiente componente, en caso de diferir de las establecidas por defecto. El comando `i2cdetect` permite escanear el bus I2C del dispositivo y obtener las direcciones de los componentes conectados. La Figura H.28 muestra la salida tras la ejecución del comando donde se observan las dos direcciones buscadas (`0x3f` y `0x38`). En el caso de que ningún componente estuviese conectado, ninguna dirección sería mostrada.

Figura H.28: Escanear bus I2C del dispositivo.

```
pi@raspberrypi:~/Desktop/hyot_raspberrypi/setup $ i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- 38 -- -- -- -- -- -- 3f
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

H.2. Componente de monitorización de sucesos del entorno

Este componente contiene la funcionalidad principal del proyecto Hyot puesto que permite la monitorización de sucesos del entorno IoT -a través del dispositivo RPi y una serie de sensores- y la generación ante un suceso anómalo de una evidencia garante de su veracidad y originación en un instante de tiempo dado. Su uso es de obligatoriedad y requiere de la ejecución previa del componente de configuración del dispositivo RPi o por el contrario la ejecución de forma manual de las acciones contenidas[2], del prototipo correctamente conectado y funcional, de las direcciones I2C de los dos LCDs conectados, de conectividad a la red en el dispositivo RPi y de la existencia del fichero de configuración `hyot.yml` en el directorio `hyot_raspberrypi/hyot/conf`.

Para iniciar este componente, desarrollado en el lenguaje Python, basta con ejecutar el fichero `hyot_main.py` -ubicado en el directorio `Hyot/hyot_raspberrypi/hyot` según la jerarquía completa del proyecto- en la RPi desde un terminal con el comando `sudo python` y seguir los pasos indicados. La monitorización del entorno se compone de tres fases ejecutadas en el siguiente orden:

[2]Las librerías requeridas se encuentran definidas en un fichero de texto (`requirements.txt`) y pueden ser instaladas ejecutando el comando: `pip install -r requirements.txt`. Si alguna librería requerida no se encuentra instalada, se muestra un error al inicio de la ejecución al intentar importarla. En el caso de la habilitación de interfaces, el comando `raspi-config` debe ser utilizado.

1. Comprobaciones iniciales.
2. Inicialización de servicios y utilidades.
3. Monitorización de sucesos del entorno.

Además, este componente ofrece un menú de ayuda, opción `-h` o `--help`, donde se detallan todas las opciones disponibles (Figura H.29), siendo las definidas las mostradas a continuación:

- Grupo general:
  - Opción `-e` o `--email`: dirección de correo electrónico destino para enviar notificaciones de alerta y de error en el proceso de medición. Si no se especifica una dirección, por defecto la funcionalidad de notificación se encuentra desactivada.
  - Opción `-m` o `--maxdistancehcsr`: distancia máxima[3] a ser medida por el sensor HC-SR04. Por defecto, 150 centímetros.
  - Opción `-r` o `--recordingtime`: tiempo de grabación del entorno cuando el protocolo de alerta es activado. Por defecto, 10 segundos.
  - Opción `-wt` o `--waittime`: tiempo de espera entre una medición y la siguiente -siempre y cuando no se active el protocolo de alerta-. Por defecto, 3 segundos.
- Grupo de pines[4]:
  - Opción `-dd` o `--dht11data`: pin de datos para el sensor DHT11. Por defecto, el pin 21 (GPIO21 -*General Purpose Input Output*-).
  - Opción `-hce` o `--hcsrecho`: pin *Echo* para el sensor HC-SR04. Por defecto, el pin 19 (GPIO19).
  - Opción `-hct` o `--hcsrtrigger`: pin *Trigger* para el sensor HC-SR04. Por defecto, el pin 26 (GPIO26).
  - Opción `-lp` o `--ledpin`: pin para el componente electrónico LED (*Light-Emitting Diode*). Por defecto, el pin 13 (GPIO13).
- Grupo de I2C:
  - Opción `-die` o `--dhti2cexpander`: tipo de expansor I2C -`MCP23008`, `MCP23017` o `PCF8574`- para el componente LCD asociado al sensor DHT11. Por defecto, `PCF8574`.
  - Opción `-dia` o `--dhti2caddress`: dirección I2C para el componente LCD asociado al sensor DHT11. Por defecto, `0x3f`.

[3]El rango de distancia máxima de medición definida en la especificación del sensor es de 4 metros. Consultar [82] para obtener más información.

[4]La especificación del número de pin debe ser en modo BCM [74].

  - Opción `-hie` o `--hcsri2cexpander`: tipo de expansor I2C -`MCP23008`, `MCP23017` o `PCF8574`- para el componente LCD asociado al sensor HC-SR04. Por defecto, `PCF8574`.
  - Opción `-hia` o `--hcsri2caddress`: dirección I2C para el componente LCD asociado al sensor HC-SR04. Por defecto, `0x38`.

- Grupo de umbrales:
  - Opción `-tt` o `--tempthreshold`: umbral límite del evento temperatura en el sensor DHT11 a partir del cual se activa el protocolo de alerta. Por defecto, 30 °C.
  - Opción `-ht` o `--humthreshold`: umbral límite del evento humedad en el sensor DHT11 a partir del cual se activa el protocolo de alerta. Por defecto, 80 %.
  - Opción `-dt` o `--distancethreshold`: umbral límite del evento distancia en el sensor HC-SR04 a partir del cual se activa el protocolo de alerta[5]. Por defecto, 50 centímetros.

Figura H.29: Monitorización de sucesos del entorno - Menú de ayuda.

```
pi@raspberrypi:~/Desktop/hyot_raspberrypi/hyot $ sudo python hyot_main.py -h
usage: hyot_main.py [-h] [-e EMAIL] [-m HCSR_MAXDISTANCE] [-r RECORDING_TIME]
                    [-wt WAITTIME_MEASUREMENT] [-dd DHT_DATAPIN]
                    [-hce HCSR_ECHOPIN] [-hct HCSR_TRIGPIN] [-lp LED_PIN]
                    [-die DHT_I2CEXPANDER] [-dia DHT_I2CADDRESS]
                    [-hie HCSR_I2CEXPANDER] [-hia HCSR_I2CADDRESS]
                    [-tt TEMPERATURE_THRESHOLD] [-ht HUMIDITY_THRESHOLD]
                    [-dt DISTANCE_THRESHOLD]

HYOT/HELP: This component monitors several events -distance,
temperature and humidity- of the environment from sensors connected to a
Raspberry Pi and in case of an anomalous reading, the alert protocol is
activated. Remember to run this component with root user or sudo and
the options are optional. If not given, default values are used.

General options:
  -h, --help            Shows the help.
  -e EMAIL, --email EMAIL
                        Email address where to send an alert notification or
                        error notification in the measurement procedure.
                        Default: disabled.
  -m HCSR_MAXDISTANCE, --maxdistancehcsr HCSR_MAXDISTANCE
                        Maximum distance to be measured by the HC-SR04 sensor
                        (e.g. 1.5 meters). Default: 1.5.
  -r RECORDING_TIME, --recordingtime RECORDING_TIME
                        Time of recording when an alert is triggered (e.g. 10
                        seconds). Default: 10.
```

[5]En este caso, el protocolo se activa cuando la distancia es menor al umbral especificado.

H.2.1. Comprobaciones iniciales

En esta primera fase se realizan diferentes comprobaciones para garantizar que la posterior ejecución se efectúe de manera correcta y sin ningún contratiempo. Si alguna de estas comprobaciones -similares a las del componente anterior- falla ya sea porque no cumple la condición o por cualquier otra causa, se muestra por consola un error al usuario y se finaliza la ejecución. Las comprobaciones incluyen:

1. Verificación de ejecución con un usuario superprivilegiado.
2. Verificación de ejecución en una plataforma GNU/Linux.
3. Verificación de ejecución en un dispositivo RPi, en cualquiera de sus versiones.
4. Verificación de conexión a Internet mediante la ejecución de un comando *ping* al buscador de Google.
5. Verificación de que no existe otra instancia de este componente en ejecución.
6. Comprobación de las opciones introducidas, donde se valida entre otros detalles:
   - El valor asociado a cada opción indicada, comprobando:
     - Opción `-e` o `--email`: dirección de correo electrónico válido (Figura H.30).
     - Opción `-r` o `--recordingtime`: tiempo de grabación del entorno mayor o igual a 1 segundo.
     - Opción `-m` o `--maxdistancehcsr`: distancia máxima mayor que 30 centímetros.
     - Opción `-wt` o `--waittime`: tiempo de espera entre una medición y la siguiente mayor o igual a 2 segundos.
     - Opciones `-dd` o `--dht11data`, `-hce` o `--hcsrecho`, `-hct` o `--hcsrtrigger` y `-lp` o `--ledpin`: número de pin GPIO entre 0-27.
     - Opciones `-die` o `--dhti2cexpander` y `-hie` o `--hcsri2cexpander`: tipo de expansor entre el rango definido: `MCP23008`, `MCP23017` o `PCF8574`.
     - Opción `-tt` o `--tempthreshold`: umbral del evento temperatura mayor o igual a 0 °C.
     - Opción `-ht` o `--humthreshold`: umbral del evento humedad entre 0-100 %.
     - Opción `-dt` o `--distancethreshold`: umbral del evento distancia entre 0-distancia máxima especificada.
   - El tipo del valor asociado a cada opción indicada (Figura H.31).
   - La validez de cada opción indicada (Figura H.32).

Figura H.30: Monitorización de sucesos del entorno - Email inválido.

```
pi@raspberrypi:~/Desktop/hyot_raspberrypi/hyot $ sudo python hyot_main.py -e email@invalid
✖ Email address entered is not a valid email. Please, type -h/--help option to get more information.
```

Figura H.31: Monitorización de sucesos del entorno - Valor asociado a la opción inválido.

```
pi@raspberrypi:~/Desktop/hyot_raspberrypi/hyot $ sudo python hyot_main.py -r dos
usage: hyot_main.py [-h] [-e EMAIL] [-m HCSR_MAXDISTANCE] [-r RECORDING_TIME]
                    [-wt WAITTIME_MEASUREMENT] [-dd DHT_DATAPIN]
                    [-hce HCSR_ECHOPIN] [-hct HCSR_TRIGPIN] [-lp LED_PIN]
                    [-die DHT_I2CEXPANDER] [-dia DHT_I2CADDRESS]
                    [-hie HCSR_I2CEXPANDER] [-hia HCSR_I2CADDRESS]
                    [-tt TEMPERATURE_THRESHOLD] [-ht HUMIDITY_THRESHOLD]
                    [-dt DISTANCE_THRESHOLD]
hyot_main.py: error: argument -r/--recordingtime: invalid int value: 'dos'
```

Figura H.32: Monitorización de sucesos del entorno - Opción inválida.

```
pi@raspberrypi:~/Desktop/hyot_raspberrypi/hyot $ sudo python hyot_main.py -a
usage: hyot_main.py [-h] [-e EMAIL] [-m HCSR_MAXDISTANCE] [-r RECORDING_TIME]
                    [-wt WAITTIME_MEASUREMENT] [-dd DHT_DATAPIN]
                    [-hce HCSR_ECHOPIN] [-hct HCSR_TRIGPIN] [-lp LED_PIN]
                    [-die DHT_I2CEXPANDER] [-dia DHT_I2CADDRESS]
                    [-hie HCSR_I2CEXPANDER] [-hia HCSR_I2CADDRESS]
                    [-tt TEMPERATURE_THRESHOLD] [-ht HUMIDITY_THRESHOLD]
                    [-dt DISTANCE_THRESHOLD]
hyot_main.py: error: unrecognized arguments: -a
```

Por último mencionar que al igual que en el componente anterior, la señal de terminación (Ctrl + C) del proceso cuando está en ejecución también es capturada, informando al usuario de que no debe finalizar la ejecución si desea continuar con la monitorización. Aun así antes de finalizar por completo la ejecución, ya sea por una señal de terminación explícita o por un error producido, se efectúa una salida ordenada liberando todos los recursos de las utilidades y servicios instanciados durante la ejecución y eliminando directorios y ficheros temporales (Figura H.33).

Figura H.33: Monitorización de sucesos del entorno - Finalización de ejecución ordenada.

```
-- Ending HYOT...

   Closing and cleaning LCD of the DHT11 sensor  ✓
   Closing and cleaning LCD of the HCSR04 sensor  ✓
   Disconnecting the Picamera  ✓
   Removing the temporary local directory: /home/pi/Desktop/hyot_raspberrypi/hyot/tempfiles  ✓
   Cleaning the GPG instance  ✓
   Disconnecting the mail session  ✓
   Disconnecting the Cloudant DB client session  ✓
   Disconnecting the Dropbox client session  ✓
```

H.2.2. Inicialización de servicios y utilidades

En esta segunda fase se inicializan y configuran todas las utilidades y servicios a emplear durante el proceso de monitorización de sucesos del entorno donde en cada acción ejecutada se comprueba si ocurre un error. En cuyo caso, se notifica al usuario y se finaliza la ejecución del componente. La inicialización incluye:

- Generar el directorio local (`rutaRepositorio/hyot/logs`) donde el fichero de *log* se almacena. Este fichero contiene la misma información que la mostrada por el terminal y su utilidad es informativa para el usuario cuando se produce un error durante alguna acción del protocolo normal de medición o protocolo de alerta de forma que se adjunta a la notificación enviada (en caso de estar activada la opción `-e` o `--email`).

- Inicializar los dos LCDs, cada uno asociado a un sensor. La Figura H.34 detallada el error mostrado en caso de que algún componente de este tipo no esté correctamente conectado, se haya indicado una dirección I2C errónea o la interfaz I2C no se encuentre habilitada. En el caso de indicar un tipo de expansor diferente al correcto, el LCD simplemente no se inicializa y por tanto no funciona pero no produce un error con lo que la ejecución continúa.

Figura H.34: Monitorización de sucesos del entorno - Error al inicializar los componentes LCDs.

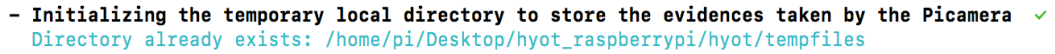
```
[pi@raspberrypi:~/Desktop/hyot_raspberrypi/hyot $ sudo python hyot_main.py -dia 099

 _    ___     ______ _______
| |  | \ \   / / __ \__   __|
| |__| |\ \_/ / |  | | | |
|  __  | \   /| |  | | | |
| |  | |  | | | |__| | | |
|_|  |_|  |_|  \____/  |_|

A PoC for traceability in IoT environments through Hyperledger Fabric by:

- Jesús Iglesias García, jesusgiglesias@gmail.com

-------------------------------------------------------

HYOT - TRACEABILITY IN IoT

This component monitors several events -distance, temperature and humidity- of the environment from
sensors connected to a Raspberry Pi and in case of an anomalous reading, the alert protocol is activated.

Type the '-h' or '--help' option to get more information.


-- Initializing HYOT...

   - Initializing the LCDs  ✖
     Error to initialize the LCDs: [Errno 22] Invalid argument. Main errno:
       - Errno 2: I2C interface is disabled.
       - Errno 22: I2C address is invalid.
       - Errno 121: LCD is not connected.

-- Ending HYOT...
```

- Inicializar la cámara. La Figura H.35 hace referencia al error mostrado en caso de que el componente *Picamera* esté conectado de forma errónea o no se haya habilitado la interfaz específica.

Figura H.35: Monitorización de sucesos del entorno - Error al inicializar la cámara.

- Inicializar el directorio local donde las evidencias -vídeos capturados sobre el entorno por la cámara- son almacenadas temporalmente hasta la finalización de un protocolo de alerta activado. En el caso de que este directorio ya exista, ninguna acción es realizada (Figura H.36).

Figura H.36: Monitorización de sucesos del entorno - Inicializar directorio local.



- Inicializar la utilidad GPG (*GNU Privacy Guard*) para la encriptación y firmado de evidencias. En esta inicialización se solicita la ruta del directorio (por defecto, `/root/.gpg/`) usado por GPG para almacenar el almacén de claves y la base de datos (BBDD) de confianza. Posteriormente, se comprueba si el directorio especificado contiene alguna clave, donde:

  - En caso de no contener ninguna clave, comienza el proceso de generación (Figura H.37) donde se solicita al usuario la indicación del nombre y dirección de correo asociada a la identidad[6] y la contraseña que protege la clave privada[7]. El par de claves generado se exporta a un fichero externo con extensión `.asc` para poder ser utilizado durante la desencriptación de la evidencia, generando también un código QR asociado a esta clave[8].

[6] En el caso de no introducir ningún valor en cualquier solicitud de datos al usuario, se utiliza el valor o valores por defecto definidos en el fichero de configuración.

[7] El usuario presenta 3 intentos para introducir la contraseña la cual debe seguir un patrón concreto: poseer una longitud mínima de 8 caracteres, al menos una letra mayúscula, una letra minúscula y un número. Esta contraseña nunca se muestra en texto claro por el terminal.

[8] Ambos ficheros son renombrados con el patrón `_índice` en caso de existir ya un fichero con dicho nombre.

Figura H.37: Monitorización de sucesos del entorno - Generación del par de claves con GPG.

```
- Initializing GPG
  Enter the path of the directory used by GPG: (/root/.gpg)
  ✓ Keyrings and trust database were successfully created
  The GPG directory does not contain any GPG key. Generating a GPG key...
  Enter the real name of the user identity: (Hyot)
  Enter the email address of the user identity: (hyot.project@gmail.com)
  Enter the password for the private key:
  ✖ The password can not be empty. Please, try it again.

  Enter the password for the private key:
  Password must contain at least 1 uppercase letter.
  Enter the password for the private key:
  Confirm the password for the private key:
  ✓ GPG key created successfully with fingerprint: 6D2DE6ED8897C59EF5547D84B812A64A87736194
  Files generated and stored in: /root/.gpg
    - QR image of the fingerprint: hyot_qr.png
    - Public and private keys: hyot_keys.asc

  It's important that you remember the fingerprint and keep these files to decrypt later.
```

- En caso de contener algún par de claves, se ofrece la posibilidad de generar uno nuevo -proceso anterior- o introducir una o varias huellas digitales (*fingerprints*) en cuyo caso es obligatorio que dichos *fingerprints* se asocien a algún par de claves existente (Figura H.38). En el caso de introducir solamente un *fingerprint*, éste es el utilizado para el proceso de encriptación y firmado de la evidencia. Por contra, si se especifica más de uno la encriptación se realiza para varios destinatarios y se solicita la indicación de qué *fingerprint* utilizar para la firma, siendo el primero de ellos el establecido como por defecto. Además, la indicación de la contraseña debe ser aquella asociada al par de clave seleccionado para la firma (Figura H.39).

Figura H.38: Monitorización de sucesos del entorno - Fingerprint no asociado a ningún par de claves existente.

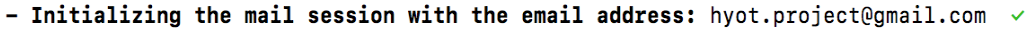
```
      - Initializing GPG
[       Enter the path of the directory used by GPG: (/root/.gpg)
        ✓ Keyrings and trust database were successfully created
        The GPG directory already contains some GPG keys
[       Please, enter the fingerprint of the key or keys (separated by commas) to use in the encryption or empty to
create a new one: 942CA41EEC10C75026E73D63A62725E30F595F11
        ✖ The fingerprint: 942CA41EEC10C75026E73D63A62725E30F595F11 does not exist in the indicated GPG directory.
```

Figura H.39: Monitorización de sucesos del entorno - Selección del fingerprint a usar para la firma.

```
      - Initializing GPG
        Enter the path of the directory used by GPG: (/root/.gpg)
        ✓ Keyrings and trust database were successfully created
        The GPG directory already contains some GPG keys
        Please, enter the fingerprint of the key or keys (separated by commas) to use in the encryption or empty to create a new one:
942CA41EEC10C75026E73D63A62725E30F595F1C,B5D6C0A3AC1546FDA32D7D9927183D31C472BB36
[       Enter the number of the fingerprint to use to sign the file [1-2]: (1) 2
[       Enter the password for the private key:
[       Confirm the password for the private key:
```

- Inicializar la sesión de *email* para las notificaciones (Figura H.40). Este proceso, que requiere de la dirección electrónica origen y credencial especificada en el fichero de configuración, solamente se efectúa si se ha indicado la opción correspondiente a la hora de ejecutar el componente.

Figura H.40: Monitorización de sucesos del entorno - Inicialización de la sesión email.



- Inicializar Cloudant NoSQL DB donde en primer lugar se solicitan las credenciales de autenticación y la dirección donde se ubica la BBDD. Posteriormente, se pregunta al usuario por el nombre de la BBDD[9] a utilizar y se comprueba si existe o no, creándola (Figura H.41) o abriéndola (Figura H.42) según el caso correspondiente. En este último caso, como la BBDD ya fue creada con anterioridad puede contener mediciones de otras ejecuciones previas.

Figura H.41: Monitorización de sucesos del entorno - Crear base de datos.

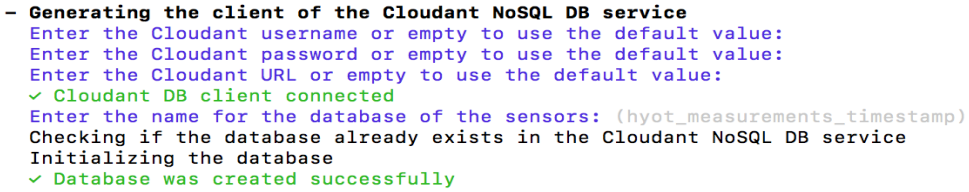
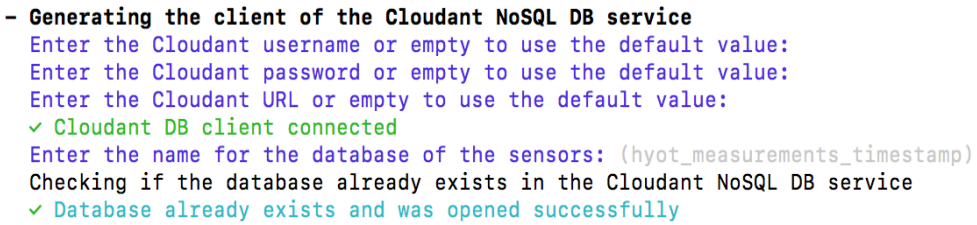


Figura H.42: Monitorización de sucesos del entorno - Instanciar base de datos ya creada.



[9]El nombre de la BBDD por defecto -`hyot_measurements_`- contiene una marca temporal o *timestamp* (año-mes) al final.

- Inicializar el servicio de almacenamiento en la nube Dropbox solicitando en primer lugar el *token* de autenticación. La Figura H.43 muestra el error cuando el dato introducido no representa un *token* mientras que la Figura H.44 muestra el error cuando éste es inválido y no pertenece a ninguna cuenta de usuario del servicio o cuando ha sido revocado.

Figura H.43: Monitorización de sucesos del entorno - Inicializar Dropbox con token malformado.

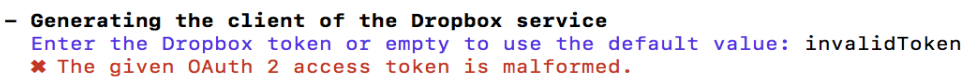
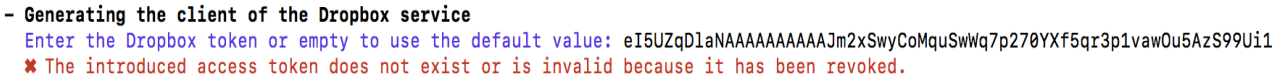


Figura H.44: Monitorización de sucesos del entorno - Inicializar Dropbox con token inválido o revocado.



Posteriormente, se comprueba si la cuenta de usuario a utilizar dispone de una cantidad mínima de espacio (500 MB) ya que en caso contrario la monitorización al no presentar suficiente espacio disponible para subir las evidencias puede verse afectada (Figura H.45). A continuación, se solicitan los nombres de los subdirectorios asociados a cada sensor (por defecto, `dht11` y `hcsr04`) donde cada uno de ellos almacenará las evidencias generadas por el sensor asociado. Los subdirectorios son creados bajo el directorio `Hyot` solamente en caso de no existir (Figura H.46).

Figura H.45: Monitorización de sucesos del entorno - Advertencia de espacio insuficiente en el servicio Dropbox.

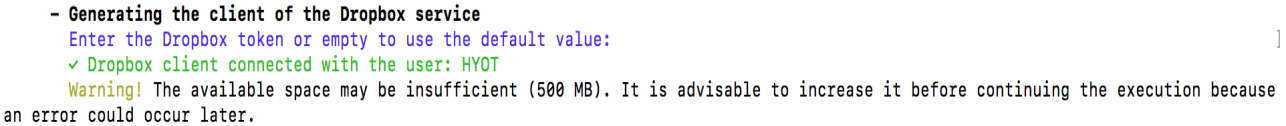


Figura H.46: Monitorización de sucesos del entorno - Directorio y subdirectorios ya existen en el servicio Dropbox.

```
- Generating the client of the Dropbox service
  Enter the Dropbox token or empty to use the default value:
  ✓ Dropbox client connected with the user: HYOT
  Enter the name of the subdirectory where the evidences of an alarm triggered by the DHT11 sensor will be stored: (/Hyot/dht11)
  Enter the name of the subdirectory where the evidences of an alarm triggered by the HC-SR04 sensor will be stored: (/Hyot/hcsr04)
  Checking if the main directory named Hyot exists in the root path of Dropbox
  ✓ Directory already exists
  Checking if the subdirectory of the DHT11 sensor exists within the Hyot directory in Dropbox
  ✓ Subdirectory already exists
  Checking if the subdirectory of the HCSR04 sensor exists within the Hyot directory in Dropbox
  ✓ Subdirectory already exists
```

- Comprobar si la BC de HF está disponible por medio de verificar si el servidor REST (*Representational State Transfer*) API (*Application Programming Interface*) de Hyperledger Composer (HC) se encuentra levantado. Para ello, se solicita la dirección URL (*Uniform Resource Locator*) y el puerto donde se ubica con el fin de efectuar un comando *ping* y verificar si efectivamente el servidor responde y ese destino en concreto se corresponde con una red de negocio. La Figura H.47 muestra el error cuando el servidor no se encuentra disponible mientras que la Figura H.48 se corresponde al error mostrado cuando el servidor está disponible en esa dirección pero no despliega una red de negocio de HC. Por su parte, la Figura H.49 muestra una comprobación correcta de este servicio.

Figura H.47: Monitorización de sucesos del entorno - Servidor HC no disponible.

```
- Checking if the business network in Hyperledger Fabric is alive
  Do you want to generate an API KEY for Composer REST server? (Y/n)
  Please, run the Composer REST server with the following API key: R3BYuK1Il6OGexhI4B74wU9vOwXXA3ozlyeWGNqyZDg
  Enter the host (e.g. IP or ngrok address) where Hyperledger Composer REST server is running: (40.121.15.202)
  Enter the port where Hyperledger Composer REST server is running: (3000)
  Checking if this host and port is alive and listen
  ✖ Connection closed in this address. Error on connect: 111.
```

Figura H.48: Monitorización de sucesos del entorno - Servidor HC disponible pero no despliega una red de negocio.

```
      - Checking if the business network in Hyperledger Fabric is alive
        Do you want to generate an API KEY for Composer REST server? (Y/n)
        Please, run the Composer REST server with the following API key: fctL-3H5h9YUQltKYzKBkSONfOQ4kjFwWbNyT_ZVMG8
[       Enter the host (e.g. IP or ngrok address) where Hyperledger Composer REST server is running: (40.121.15.202)
        Enter the port where Hyperledger Composer REST server is running: (3000)
[       Checking if this host and port is alive and listen
[       ✓ Port 3000 in 40.121.15.202 reachable
        Pinging the business network of the address: https://40.121.15.202:3000
        ✖ Participant key in the response does not contain the following expressions: org.hyperledger.composer.system.NetworkAdmin#
or org.hyot.network.User#.
```

Figura H.49: Monitorización de sucesos del entorno - Comprobación de servidor HC correcta.

```
      - Checking if the business network in Hyperledger Fabric is alive
[       Do you want to generate an API KEY for Composer REST server? (Y/n)
        Please, run the Composer REST server with the following API key: s0ugGl3xTgvk-HvhMQ7yASmICznUQkS4tgRItdPRsF4
[       Enter the host (e.g. IP or ngrok address) where Hyperledger Composer REST server is running: (40.121.15.202)
[       Enter the port where Hyperledger Composer REST server is running: (3000)
        Checking if this host and port is alive and listen
        ✓ Port 3000 in 40.121.15.202 reachable
        Pinging the business network of the address: https://40.121.15.202:3000
        ✓ Business network is alive in the address
[       Enter the username for the owner of the alerts: (hyotRPi) hyotUser
[       Enter the email for the owner of the alerts: (optional)
[       Enter the first name for the owner of the alerts: (Hyot)
[       Enter the last name for the owner of the alerts: (Raspberry Pi)
        Creating a new User participant in the Blockchain of Hyperledger Fabric
        ✓ New participant created successfully: hyotUser. Use this username in the web system or to register a normal user
in this application.
```

Además, se ofrece al usuario la posibilidad de introducir la información correspondiente al participante de tipo `User` de la BC que se va a encargar de registrar las alertas (activos `Alert`) en dicha capa de persistencia para la monitorización actual. De tal manera, que si el nombre de usuario indicado ya está asociado a un participante existente ninguna acción es realizada, como se muestra en la Figura H.50. Por contra, si no existe un participante con ese nombre de usuario, se procede a su creación como se muestra en la figura anterior y será este participante el poseedor de cualquier alerta registrada en dicha monitorización.

Figura H.50: Monitorización de sucesos del entorno - Participante existente en la BC.

```
- Checking if the business network in Hyperledger Fabric is alive
  Do you want to generate an API KEY for Composer REST server? (Y/n)
  Please, run the Composer REST server with the following API key: A-We3S3YrA7h5RioZmjQsAEhu_AknY0WL-IHvXUWt38
  Enter the host (e.g. IP or ngrok address) where Hyperledger Composer REST server is running: (40.121.15.202)
  Enter the port where Hyperledger Composer REST server is running: (3000)
  Checking if this host and port is alive and listen
  ✓ Port 3000 in 40.121.15.202 reachable
  Pinging the business network of the address: https://40.121.15.202:3000
  ✓ Business network is alive in the address
  Enter the username for the owner of the alerts: (hyotRPi)
  Participant already exists: hyotRPi. Use this username in the web system or to register a normal user in this application.
```

También, con el fin de proporcionar una capa de seguridad extra se ofrece la funcionalidad de securizar el servidor de HC mediante un *api-key* (por defecto, generado) el cual es introducido en cada petición transmitida[10]. En caso de securizar el servidor pero no enviar el *api-key* en la petición o enviar uno diferente, ésta es denegada, tal y como muestra la Figura H.51.

[10]En el caso de no hacer uso de esta opción, el servidor de HC puede ser arrancado con anterioridad. En caso contrario, se debe obtener el *api-key* en primer lugar y después arrancar el servidor indicando este valor en la opción correspondiente.

Figura H.51: Monitorización de sucesos del entorno - Petición denegada con servidor HC securizado.

```
- Checking if the business network in Hyperledger Fabric is alive
  Do you want to generate an API KEY for Composer REST server? (Y/n) n
  Enter the host (e.g. IP or ngrok address) where Hyperledger Composer REST server is running: (40.121.15.202)
  Enter the port where Hyperledger Composer REST server is running: (3000)
  Checking if this host and port is alive and listen
  ✓ Port 3000 in 40.121.15.202 reachable
  Pinging the business network of the address: https://40.121.15.202:3000
  ✖ Error 401: Unauthorized request. Please, enter a valid API key to submit the request to the Hyperledger Composer REST server.
```

Hay que tener en cuenta que este servidor también requiere de securización mediante HTTPS (*Hypertext Transfer Protocol Secure*) -peticiones son efectuadas únicamente a través de este protocolo- por lo que es necesario generar un certificado[11].

H.2.3. Monitorización de sucesos del entorno

Esta fase, de ejecución continúa hasta que sea parada explícitamente por el usuario o se produzca un error, permite monitorizar los sucesos que se producen en el entorno IoT en busca de garantizar la veracidad e integridad del hecho de un suceso anómalo. Cada medición conformada entre otra información por un identificador único, un *timestamp* y valores de los eventos: temperatura, humedad y distancia, es obtenida en situación normal cada 3 segundos como valor por defecto y es filtrada para determinar si presenta un suceso anómalo o no en base a los umbrales límite indicados para cada evento. En base a esta información:

- La medición actual puede no presentar un caso extraño (Figura H.52), activando el protocolo normal de medición consistente en:

  1. Almacenar la medición en la BBDD para que quede constancia de que en ese momento temporal el entorno controlado no presenta ninguna incidencia.

[11] Para el contexto actual se ha generado un certificado autofirmado lo que implica también la indicación de ello en el fichero de configuración.

Figura H.52: Monitorización de sucesos del entorno - Medición normal.

```
-- Information - Values established:
   -- Alert thresholds:
     - Sensor: DHT11   | Event: Temperature | > 30 °C
     - Sensor: DHT11   | Event: Humidity    | > 80 %
     - Sensor: HC-SR04 | Event: Distance    | < 50 cm
   -- Recording time: 10 seconds
   -- Time between measurements: 3 seconds
   -- Maximum distance to be measured (HC-SR04): 1.5 meters

-- Reading values each 3 seconds from sensors

Measurement 1
UUID: bb71c254-f54f-4a2e-8d74-565095b145f0
Datetime: 29-08-2018 19:57:18 PM
DHT11 sensor
Temperature:  28.0 °C
Humidity:  45.0 %
HC-SR04 sensor
Distance:  150.000 cm

  -- Adding the measurement to the database: hyot_measurements_2018-08  ✓

------------------------------

Measurement 2
UUID: 704ddd23-75a9-4f18-a0a1-c0f7bf15443b
Datetime: 29-08-2018 19:57:22 PM
DHT11 sensor
Temperature:  28.0 °C
Humidity:  48.0 %
HC-SR04 sensor
Distance:  150.000 cm

  -- Adding the measurement to the database: hyot_measurements_2018-08  ✓
```

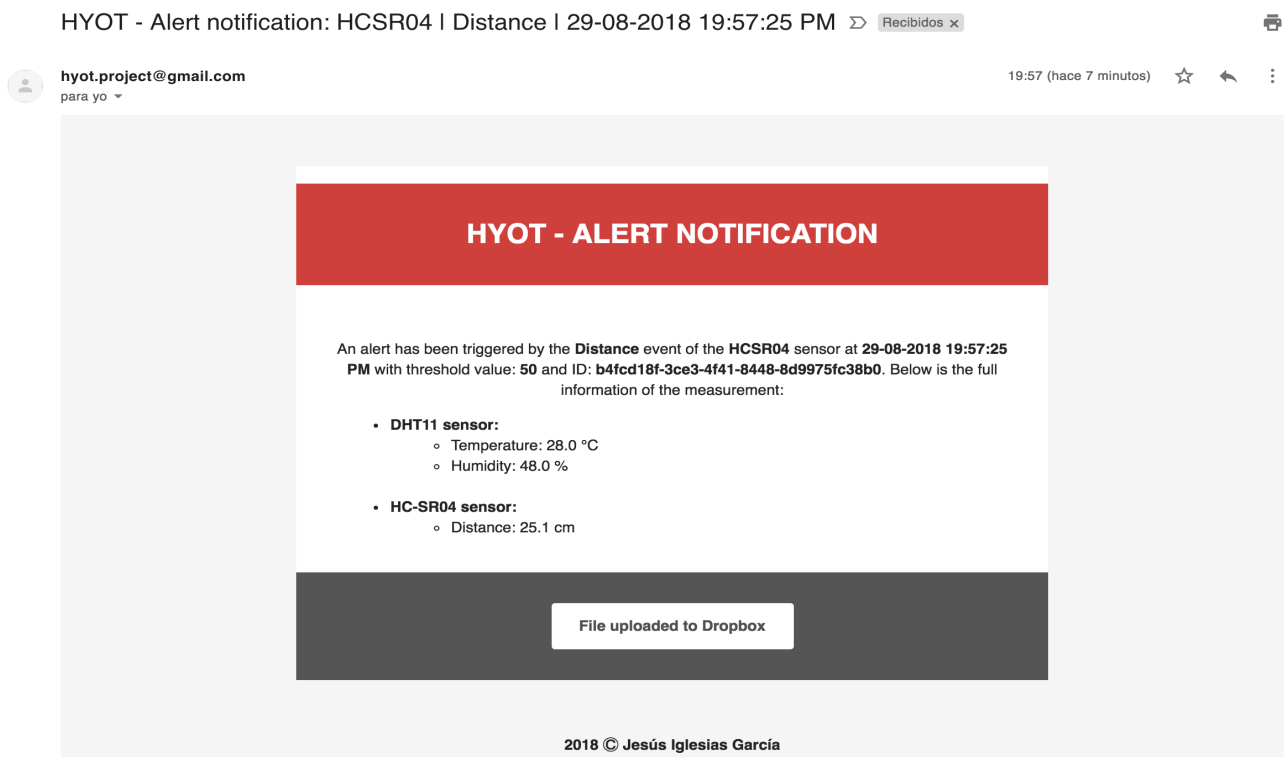
- La medición actual puede presentar un caso extraño (Figura H.53), activando el protocolo de alerta donde se indica el sensor y evento originario. Este protocolo consiste en:

  1. Captura de un vídeo -por defecto 10 segundos de duración- a través de una cámara (dispositivo Picamera) conectada a la RPi el cual representa la evidencia.
  2. Cálculo del valor *hash* del contenido de la evidencia. Este valor actúa como prueba de integridad y se almacena únicamente en la BC.
  3. Encriptación y firmado de la evidencia con la utilidad GPG.
  4. Almacenamiento de la evidencia encriptada y firmada en el servicio de almacenamiento en la nube, Dropbox.
  5. Almacenamiento de la medición con información adicional en la BBDD para que quede constancia de que en ese momento temporal el entorno vigilado presentó una lectura anómala.
  6. Registro de la incidencia en la BC de HF con la generación de un activo *Alert*.
  7. Notificación del suceso no controlado mediante el envío de un *email* a la dirección de correo electrónico destino configurada en caso de estar activado (Figura H.54).

Figura H.53: Monitorización de sucesos del entorno - Medición anómala.

```
Measurement 3
UUID: b4fcd18f-3ce3-4f41-8448-8d9975fc38b0
Datetime: 29-08-2018 19:57:25 PM
DHT11 sensor
Temperature:  28.0 °C
Humidity:  48.0 %
HC-SR04 sensor
Distance:  25.100 cm

---------- ALERT TRIGGERED | HCSR04 | Distance ----------
---------- Initiating the alert protocol ------------
  -- Taking a recording of 10 seconds and temporarily storing it in the path: /home/pi/Desktop/hyot_raspberrypi/hyot/tempfiles/hcsr04_distance
_29082018_195725.h264  ✓
  -- Applying a hash function to the content of the evidence  ✓
  -- Encrypting and signing the evidence  ✓
  -- Uploading the evidence to Dropbox to the path: /Hyot/hcsr04/hcsr04_distance_29082018_195725.h264.gpg  ✓
  -- Adding the measurement to the database: hyot_measurements_2018-08  ✓
  -- Submitting the transaction to publish the alert to the Blockchain of Hyperledger Fabric  ✓
  -- Sending alert notification to the following email address: jesusgiglesias@gmail.com  ✓
  -- Removing the temporary local evidence: /home/pi/Desktop/hyot_raspberrypi/hyot/tempfiles/hcsr04_distance_29082018_195725.h264  ✓
  -- Removing the encrypted and signed evidence: /home/pi/Desktop/hyot_raspberrypi/hyot/tempfiles/hcsr04_distance_29082018_195725.h264.gpg  ✓
---------- PROTOCOL FINISHED. CONTINUING... ----------
```

Figura H.54: Monitorización de sucesos del entorno - Correo electrónico notificando un suceso anómalo.



Además, en el caso de que algún valor obtenido sea inválido únicamente se comprueba el evento que contenga el valor válido siguiendo el procedimiento anterior (Figura H.55). Por el contrario, si ambos valores son inválidos esa medición se omite y se continúa la monitorización.

Figura H.55: Monitorización de sucesos del entorno - Medición con eventos del sensor DHT11 inválidos.

```
Measurement 1
UUID: bcc5d162-e208-49e0-84e1-58c3ef078a98
Datetime: 29-08-2018 20:09:36 PM
DHT11 sensor
Failed to get reading. Humidity and temperature are invalid or None.
HC-SR04 sensor
Distance:  150.000 cm

  -- Adding the measurement to the database: hyot_measurements_2018-08  ✓
```

Si durante alguna acción del protocolo normal de medición o protocolo de alerta se produce un error, la ejecución es finalizada enviando una notificación *email* (si la opción correspondiente está activada) y adjuntado el fichero de *log* (Figura H.56) o notificando que es recomendable activar dicha opción para conseguir el aviso instantáneo y la traza (Figura H.57). La Figura H.58 muestra el correo electrónico recibido con diseño adaptativo para una notificación *email* de este tipo.

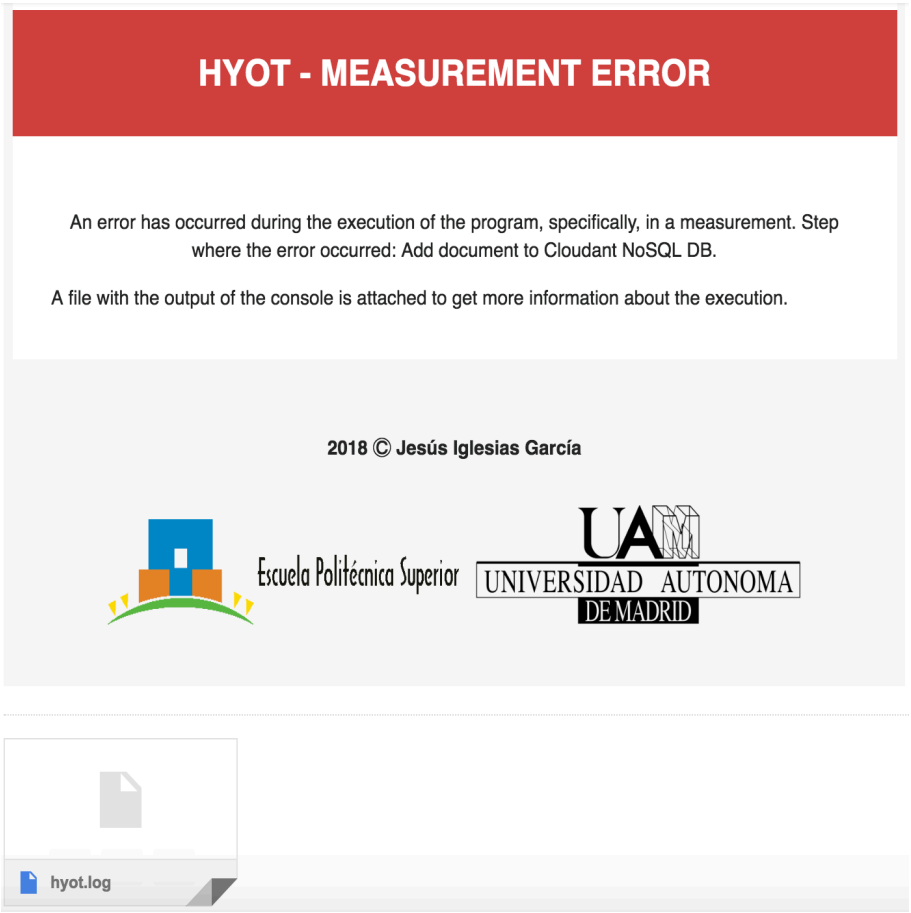
Figura H.56: Monitorización de sucesos del entorno - Error y notificación durante una acción del protocolo normal de medición.

```
Measurement 1
UUID: 2662c7ed-b85e-40ac-89cd-4ca5c91b43aa
Datetime: 29-08-2018 20:18:51 PM
DHT11 sensor
Temperature:  28.0 °C
Humidity:  49.0 %
HC-SR04 sensor
Distance:  150.000 cm

  -- Adding the measurement to the database: hyot_measurements_2018-08  ✖ Error to add the measurement. A measurement
with the same identifier already exists. Please, check the Cloudant NoSQL DB service.

  Aborting the execution...
  Sending email to the following email address: jesusgiglesias@gmail.com to notify the measurement error ✓
```

Figura H.57: Monitorización de sucesos del entorno - Error durante una acción del protocolo normal de medición.

```
Measurement 1
UUID: 73c20d93-96d3-40b6-963b-a01b1cb61557
Datetime: 29-08-2018 20:20:01 PM
DHT11 sensor
Temperature:  28.0 °C
Humidity:  50.0 %
HC-SR04 sensor
Distance:  150.000 cm

  -- Adding the measurement to the database: hyot_measurements_2018-08  ✖ Error to add the measurement. A measurement with
the same identifier already exists. Please, check the Cloudant NoSQL DB service.

  Aborting the execution...
  Information! Consider enabling the error notification by email to receive an informative email instantly. To do this, use
the -e/--email option or type -h/--help option to get more information.
```

Figura H.58: Monitorización de sucesos del entorno - Correo electrónico notificando un error durante alguna acción del protocolo de medición.



H.3. Componente de desencriptación de evidencia

Este componente del proyecto Hyot permite efectuar la desencriptación, la verificación de firma y la comprobación de integridad de contenido de una evidencia previamente encriptada y firmada con la herramienta GPG. Su uso, aunque es totalmente opcional puesto que los pasos que son efectuados pueden ser ejecutados de forma manual ya sea a través de la línea de comandos o con alguna herramienta externa, requiere de la instalación de Python y de la utilidad GPG [24] en la máquina donde se ejecutará el componente, además de una serie de librerías Python. Esta configuración puede ser omitida si se ha ejecutado en la RPi el componente de configuración y si el componente de desencriptación se va a ejecutar en ese mismo dispositivo lo cual es lo recomendable. En otro caso, requiere de las dependencias mencionadas[12].

[12]Las librerías se encuentran definidas en un fichero de texto (`requirements.txt`) y pueden ser instaladas ejecutando el comando: `pip install -r requirements.txt`. Si alguna librería requerida no se encuentra instalada, se muestra un error al inicio de la ejecución al intentar importarla.

Para iniciar este componente basta con ejecutar el fichero `hyot_decryption.py` -ubicado en el directorio `Hyot/hyot_raspberrypi/hyot_decryption` según la jerarquía completa del proyecto- en la RPi o en cualquier otro dispositivo electrónico –siempre y cuando disponga de las dependencias indicadas anteriormente- desde un terminal con el comando `sudo python`.

Este componente está compuesto de dos fases ejecutadas en el siguiente orden:

1. Comprobaciones iniciales.
2. Proceso completo de desencriptación.

Además, este componente ofrece un menú de ayuda, opción `-h` o `--help`, donde se detallan todas las opciones disponibles (Figura H.59), siendo las definidas las mostradas a continuación:

- Opción `-g` o `--gpghome` (obligatoria): directorio donde GPG generará su almacén (base de datos de confianza y anillos de clave pública y privada).
- Opción `-e` o `--encryptedfile`: ruta en el sistema local donde la evidencia se encuentra.
- Opción `-l` o `--link`: enlace a la nube donde la evidencia se encuentra.
- Opción `-ha` o `--hash` (obligatoria): valor *hash* de la evidencia original sin encriptar ni firmar. Este dato se obtiene del mecanismo de persistencia BC.
- Opción `-f` o `--fingerprint`: huella digital o *fingeprint* asociado al par de claves a utilizar.
- Opción `-k` o `--keys`: ruta en el sistema local del fichero que contiene el par de claves.
- Opción `-d` o `--decryptedhome`: directorio donde la evidencia desencriptada será almacenada. Por defecto, misma ruta que aquella donde se almacena la evidencia encriptada y firmada.

Figura H.59: Desencriptación de evidencia - Menú de ayuda.

```
pi@raspberrypi:~/Desktop/hyot_raspberrypi/hyot_decryption $ sudo python hyot_decryption.py --help
usage: hyot_decryption.py [-h] -g GPGHOME [-e ENCRYPTEDFILE] [-l LINK] -ha
                          HASHFILE [-f FINGERPRINT] [-k KEYS]
                          [-d DECRYPTEDHOME]

HYOT DECRYPTION/HELP: This component allows to decrypt an evidence
previously encrypted with GPG, verify the sign and the integrity of the
content.

General options:
  -h, --help            Shows the help.
  -g GPGHOME, --gpghome GPGHOME
                        Directory where the public and private key rings and
                        the trust database of GPG are located.
  -e ENCRYPTEDFILE, --encryptedfile ENCRYPTEDFILE
                        Local path of the encrypted and signed evidence with
                        GPG (optional).
  -l LINK, --link LINK  Link where the encrypted and signed evidence is stored
                        in the Cloud (optional).
  -ha HASHFILE, --hash HASHFILE
                        Hash code of the content of the original evidence
                        (decrypted).
  -f FINGERPRINT, --fingerprint FINGERPRINT
                        Fingerprint of the pair of keys to use (optional).
  -k KEYS, --keys KEYS  Path of the file which stores the public and private
                        key (optional).
  -d DECRYPTEDHOME, --decryptedhome DECRYPTEDHOME
                        Directory where the decrypted evidence will be store
                        (optional). Default: same directory that the encrypted
                        evidence when -e/--encryptedfile option is introduced.
```

H.3.1. Comprobaciones iniciales

En esta primera fase se realizan las siguientes dos comprobaciones, finalizando la ejecución y notificando al usuario del error en caso de que alguna condición no se cumpla.

1. Verificación de ejecución con un usuario superprivilegiado (Figura H.60).

Figura H.60: Desencriptación de evidencia - Verificación de usuario.

```
pi@raspberrypi:~/Desktop/hyot_raspberrypi/hyot_decryption $ python hyot_decryption.py
✖ You need to have root privileges to run this component. Please, try it again using sudo.
```

2. Comprobación de las opciones introducidas, donde se valida entre otros detalles:

   - La introducción de las opciones obligatorias (Figuras H.61 y H.62).

Figura H.61: Desencriptación de evidencia - Verificación de opción obligatoria -g o –gpghome.

```
pi@raspberrypi:~/Desktop/hyot_raspberrypi/hyot_decryption $ sudo python hyot_decryption.py
usage: hyot_decryption.py [-h] -g GPGHOME [-e ENCRYPTEDFILE] [-l LINK] -ha
                          HASHFILE [-f FINGERPRINT] [-k KEYS]
                          [-d DECRYPTEDHOME]
hyot_decryption.py: error: argument -g/--gpghome is required
```

Figura H.62: Desencriptación de evidencia - Verificación de opción obligatoria -ha o –hash.

```
[pi@raspberrypi:~/Desktop/hyot_raspberrypi/hyot_decryption $ sudo python hyot_decryption.py -g /home/pi/
usage: hyot_decryption.py [-h] -g GPGHOME [-e ENCRYPTEDFILE] [-l LINK] -ha
                          HASHFILE [-f FINGERPRINT] [-k KEYS]
                          [-d DECRYPTEDHOME]
hyot_decryption.py: error: argument -ha/--hash is required
```

- La introducción de un método para indicar la evidencia a utilizar (Figuras H.63 y H.64).

Figura H.63: Desencriptación de evidencia - Verificación de indicación de evidencia.

```
[pi@raspberrypi:~/Desktop/hyot_raspberrypi/hyot_decryption $ sudo python hyot_decryption.py -g /home/pi/ -ha dde3ad060bf]
a61ef708ec4e0fba72c980c075a1e498df5cf3834a6eb1af6b76137831858be6c55ba63ea4523777891b6d87ad10cf73d42e2ab0ece196f2e4879
✖ Please, enter some method to indicate the evidence to use or type -h/--help option to get more information.
```

Figura H.64: Desencriptación de evidencia - Otra verificación de indicación de evidencia.

```
[pi@raspberrypi:~/Desktop/hyot_raspberrypi/hyot_decryption $ sudo python hyot_decryption.py -g /home/pi/ -ha dde3ad060bfa61ef708ec4e0]
fba72c980c075a1e498df5cf3834a6eb1af6b76137831858be6c55ba63ea4523777891b6d87ad10cf73d42e2ab0ece196f2e4879 -e /home/pi/Desktop/hcsr04_
distance_02082018_183432.h264.gpg -l https://www.dropbox.com/s/j43evlgrdv6z3mj/hcsr04_distance_02082018_183432.h264.gpg?dl=0
✖ Please, enter only one way to indicate the evidence to use (local file or link) or type -h/--help option to get more information.
```

- La introducción de un método para indicar el par de claves a utilizar -caso similar al anterior con respecto a las comprobaciones efectuadas-.
- La validez de cada opción indicada.

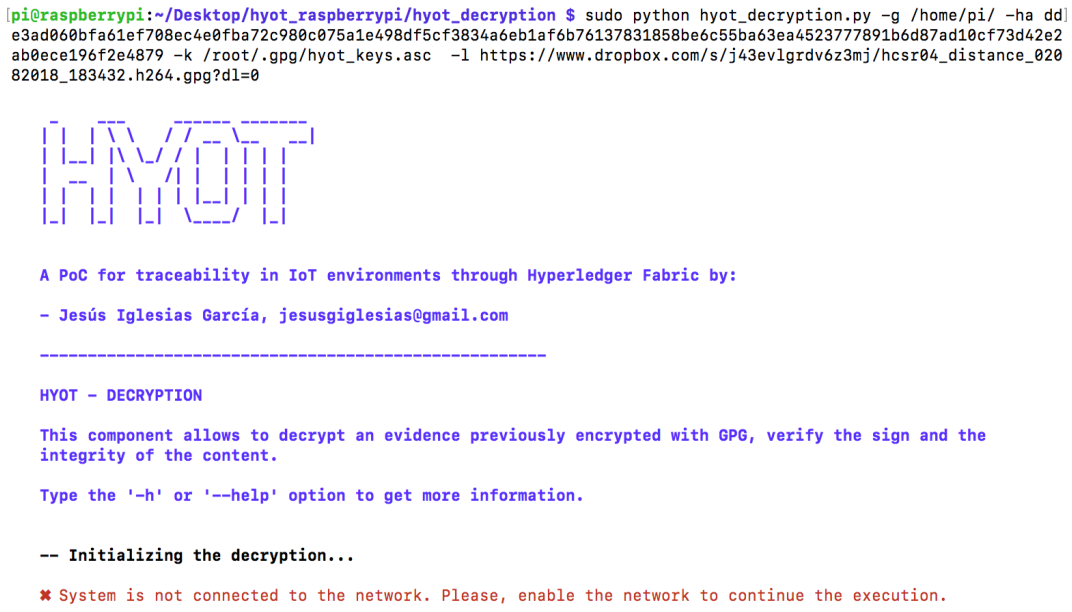
Por último, mencionar que al igual que en el resto de componentes la señal de terminación (Ctrl + C) del proceso cuando está en ejecución también es capturada, informando al usuario de que debería esperar hasta que la ejecución finalizase de forma autónoma.

H.3.2. Proceso completo de desencriptación

Esta segunda fase contiene la funcionalidad núcleo de este componente donde se realizan las acciones necesarias para verificar la autenticidad e integridad de la evidencia que actuará como prueba irrefutable de que un suceso anómalo se produjo en un instante temporal dado. En primer lugar, si el método de indicación de la evidencia a utilizar es a través del enlace al servicio de almacenamiento en la nube donde se encuentra (opción `-l` o `--link`), se procede a:

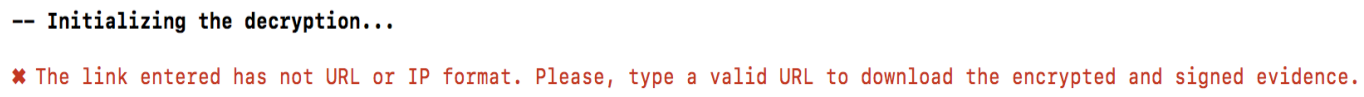
- Verificar si el dispositivo se encuentra conectado a la red mediante la ejecución de un *ping* al buscador de Google (Figura H.65).

Figura H.65: Desencriptación de evidencia - Verificación de conexión a red.



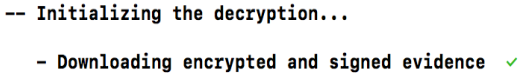
- Validar el enlace introducido (Figura H.66).

Figura H.66: Desencriptación de evidencia - Validar enlace.



- Descargar y almacenar la evidencia encriptada y firmada en el directorio del componente[13] (Figura H.67). En el caso de producirse algún error durante la descarga o almacenamiento de la evidencia descargada, se notifica al usuario y se finaliza la ejecución. La Figura H.68 muestra un ejemplo de error durante la descarga.

Figura H.67: Desencriptación de evidencia - Descargar evidencia.



[13]El código está adaptado al enlace compartido proporcionado por el servicio Dropbox el cual corresponde con el almacenado en los mecanismos de persistencia. El uso de otros servicios o formatos de enlace podría alterar el funcionamiento normal de este componente.

Figura H.68: Desencriptación de evidencia - Error al descargar la evidencia.

```
-- Initializing the decryption...
   - Downloading encrypted and signed evidence  ×
   ✖ Error to download the evidence.
```

Los pasos arriba mencionados son omitidos si la indicación de la evidencia a utilizar se hace a través de la ruta en el sistema local donde se encuentra (opción `-e` o `--encryptedfile`). Para ambos casos, la siguiente comprobación consiste en validar la existencia de los directorios y ficheros indicados en las opciones a la hora de la ejecución, es decir, se valida que el directorio GPG, el fichero del par de claves, la evidencia encriptada y firmada y el directorio de desencriptación existan en el sistema local y sean del tipo correcto, es decir, un directorio o un fichero según corresponda. La Figura H.69 muestra un ejemplo donde se ha indicado una ruta de evidencia inexistente.

Figura H.69: Desencriptación de evidencia - Evidencia inexistente.

```
-- Initializing the decryption...
✖ The encrypted and signed evidence does not exist or is not a file in the local system.
```

Si esta primera comprobación es correcta, se procede a verificar que la evidencia encriptada y firmada posea la extensión adecuada, `.gpg` (Figura H.70).

Figura H.70: Desencriptación de evidencia - Evidencia con extensión incorrecta.

```
-- Initializing the decryption...
✖ The encrypted and signed evidence has an extension which is not allowed. It must be a file with format: .gpg.
```

Posteriormente, en función de la opción indicada por el usuario para indicar el par de claves a utilizar, se procede a:

- Comprobar el *fingerprint* introducido (opción `-f` o `--fingerprint`) donde se verifica que el directorio GPG indicado contenga al menos un par de claves (Figura H.71) y que el *fingerprint* pertenezca a un par de claves existente (Figura H.72). Si ambas comprobaciones son correctas, el par de claves asociado a dicho *fingerprint* será el utilizado para el proceso de desencriptación y verificación de la firma.

Figura H.71: Desencriptación de evidencia - Directorio GPG no contiene ningún par de claves.

```
-- Initializing the decryption...

  ✖ The GPG directory does not contain any public or private key. Please, import the pair of keys with
the -k/--keys option to continue the process.
```

Figura H.72: Desencriptación de evidencia - Fingerprint no pertenece a ningún par de claves existente.

```
-- Initializing the decryption...

  ✖ The entered fingerprint does not exist in the indicated GPG directory. Please, import the pair of
keys to continue the process or use an existing fingerprint.
```

- Importar el fichero que contenga el par de claves (opción `-k` o `--keys`). En caso de no contener una clave pública y privada, se muestra el error de la Figura H.73.

Figura H.73: Desencriptación de evidencia - Fichero no contiene el par de claves.

```
-- Initializing the decryption...

  ✖ The entered key file does not contain the pair of keys (public and private key). Please, use the file
generated in the component of monitoring of environmental events.
```

A partir de aquí, y una vez se ha cerciorado que todos los datos introducidos son correctos, ya comienza el proceso de desencriptación en sí donde el primer paso es solicitar al usuario la introducción de la contraseña de protección de la clave privada a usar. Este dato de entrada no puede ser vacío[14] en cuyo caso el usuario tiene 3 intentos. Agotados el límite de intentos, la ejecución finaliza (Figura H.74).

Figura H.74: Desencriptación de evidencia - Límite de intentos agotado.

```
-- Initializing the decryption...

   Enter the password for the private key:
   ✖ The password can not be empty. Please, try it again.

   Enter the password for the private key:
   ✖ The password can not be empty. Please, try it again.

   Enter the password for the private key:
   ✖ Number of attempts spent. Please, run again the code.
```

[14]Si el dato contuviese espacios al inicio o final, éstos son eliminados.

El siguiente paso efectúa la desencriptación de la evidencia utilizando la clave privada. El fichero resultante -evidencia desencriptada en formato `.h264` (Figura H.75)- se almacena en el mismo directorio donde se encuentra la evidencia encriptada y firmada o en el directorio destino si se introduce un valor para la opción `-d` o `--decryptedhome`. La Figura H.76 muestra un error durante la desencriptación donde éste puede ser debido a diferentes circunstancias tal como, contraseña de clave privada errónea, diferente par de claves empleado, etc.

Figura H.75: Desencriptación de evidencia correcta.

```
- Decrypting  ✓

Evidence successfully decrypted in the path: hcsr04_distance_02082018_183432.h264.
```

Figura H.76: Desencriptación de evidencia - Error en la desencriptación.

```
-- Initializing the decryption...
   Enter the password for the private key:
   - Decrypting  ✗
   The decryption has failed. Main reasons:
    - Evidence was encrypted with another RSA key.
    - Bad passphrase to unlock the GPG secret key.
    - No valid OpenPGP data found.
    - Output is a directory.
```

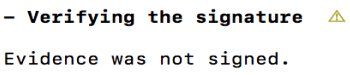
Tras desencriptar la evidencia se procede a verificar la firma empleando la clave pública, mostrando la identidad del usuario, la fecha, el *fingerprint* usado para la firma y el nivel de confianza del par de claves utilizado[15] (Figura H.77). En caso de que la evidencia no fuese firmada también se notifica al usuario (Figura H.78).

Figura H.77: Desencriptación de evidencia - Verificación de firma.

```
- Verifying the signature  ✓

Information of the signature
----------------------------
User identity: Hyot <hyot.project@gmail.com>
Signature made: 2018-08-02 18:35:04
Fingerprint used: DC84141823586C59CEFB2FD9C45908D8982B0206

Trust level of the key used
---------------------------
4 - TRUST_ULTIMATE
```

[15]En caso de importar el par de claves el nivel de confianza es bajo. Por el contrario, si se emplea un *fingerprint* y el mismo directorio GPG usado para la encriptación y firmado el nivel de confianza es alto.

Figura H.78: Desencriptación de evidencia - Evidencia no firmada.



Por último, se comprueba que el contenido de la evidencia original no haya sido alterado por un supuesto intruso no autorizado a través de comparar el valor *hash* almacenado en la BC (introducido con la opción `-ha` o `--hash`) con el valor calculado de la evidencia recién desencriptada. La Figura H.79 muestra un ejemplo donde ambos valores son iguales mientras que la Figura H.80 muestra el ejemplo contrario. Además, en el caso de producirse un error durante el cálculo de este valor también se notifica al usuario (Figura H.81).

Figura H.79: Desencriptación de evidencia - Valores hashes iguales.

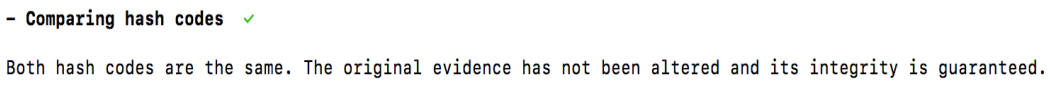


Figura H.80: Desencriptación de evidencia - Valores hashes diferentes.

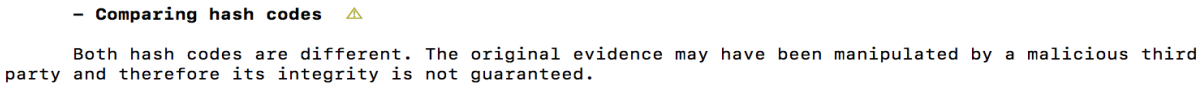


Figura H.81: Desencriptación de evidencia - Error al calcular el valor hash.

Tras ejecutar este componente, se obtiene la evidencia original la cual se corresponde con el vídeo tomado durante el protocolo de alerta de una incidencia en la monitorización de sucesos del entorno y cuyo nombre contiene el sensor y evento que lanzó la alerta y la fecha y hora cuando fue tomado. Si todas las comprobaciones son correctas, se puede garantizar la originalidad y veracidad del contenido de este vídeo en el instante de tiempo en el que se produjo.

H.4. Componente - Sistema web

Este componente del proyecto Hyot, desarrollado con el *framework* Grails y ubicado en el directorio `Hyot/hyot_app` según la jerarquía completa del proyecto, hace referencia al sistema web donde el usuario puede consultar en tiempo real la información monitorizada del entorno IoT así como los sucesos anómalos que se produzcan obteniendo a su vez la información necesaria para la obtención y verificación de la prueba irrefutable generada. Este sistema, para el despliegue actual, se encuentra localizado en la URL: `https://hyot.eu-gb.mybluemix.net` y basta con acceder a dicha dirección para poder comenzar a usar la solución web[16].

H.4.1. Página de inicio

El sistema web es una solución donde es obligatorio la autenticación, es decir, poseer una cuenta de usuario para poder acceder a la funcionalidad del sistema lo que conlleva a que un usuario no registrado solamente pueda visualizar las páginas de iniciar sesión y restablecer contraseña. Por tanto, la primera toma de contacto que presenta el usuario es la página correspondiente al inicio de sesión junto con la opción de restaurar contraseña (Figura H.82).

El inicio de sesión se presenta a través de la introducción de las credenciales, compuesta por:

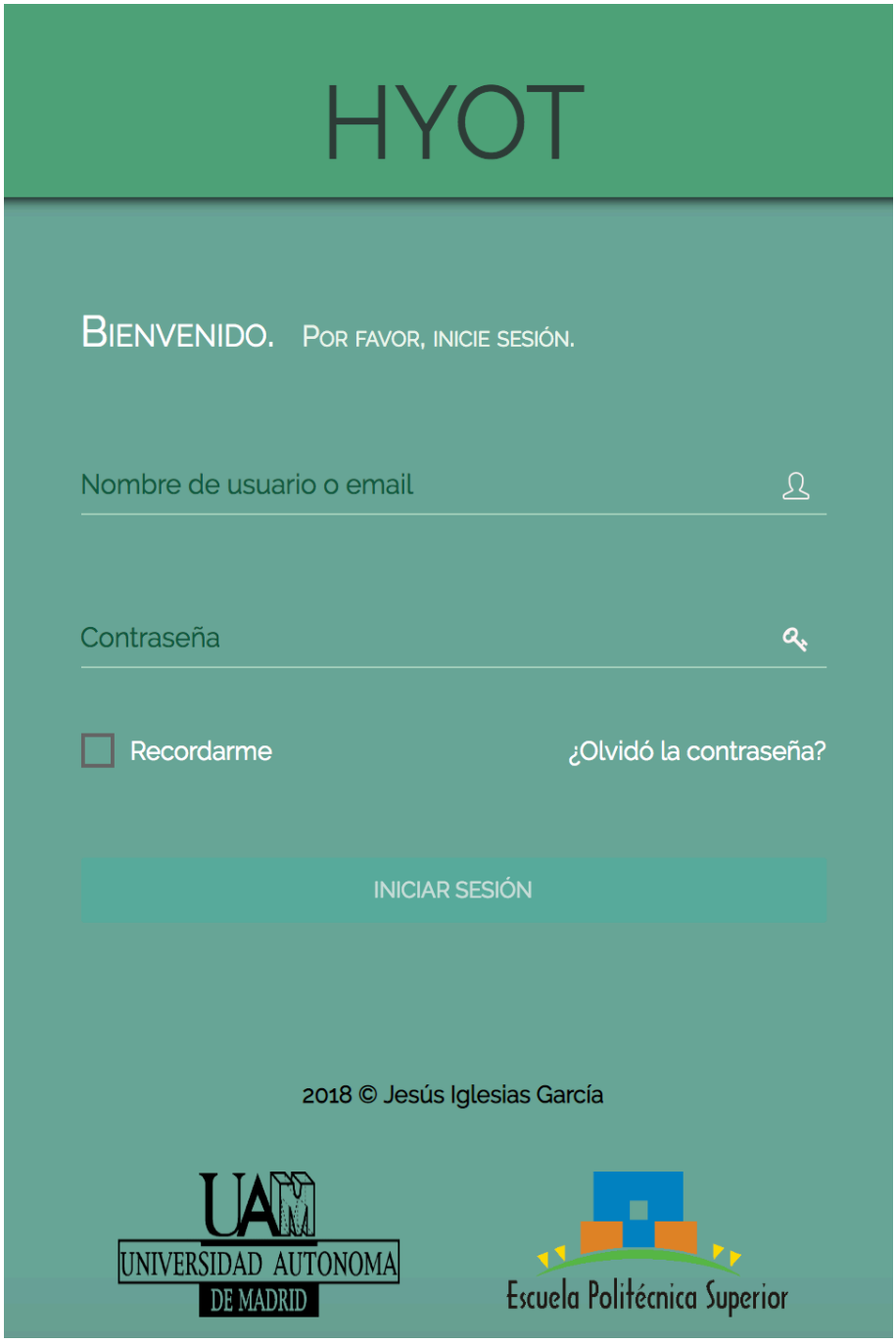
- Nombre de usuario o *email*.
- Contraseña.

siendo elección del propio usuario la introducción del nombre de usuario o *email*, ambas opciones totalmente válidas[17]. Las entradas de texto en el sistema web poseen elementos que mejoran la interacción del usuario con la interfaz como, por ejemplo el contador límite de caracteres que permite el campo de entrada, el icono aspa para borrar el texto introducido o el icono ojo para visualizar la contraseña en un campo confidencial, ambos apareciendo a la derecha de la entrada cuando se comienza a introducir el texto en él.

[16]Requiere que el servidor de HC se encuentre en ejecución para poder hacer las consultas contra la BC.

[17]En algunas casos el botón de realizar una determinada acción (p.ej. botón de iniciar sesión) se presenta como desactivado hasta que el usuario complete los campos necesarios. En ese instante, el botón se activa y se permite pulsar sobre él para efectuar la acción.

Figura H.82: Sistema web - Página de inicio.

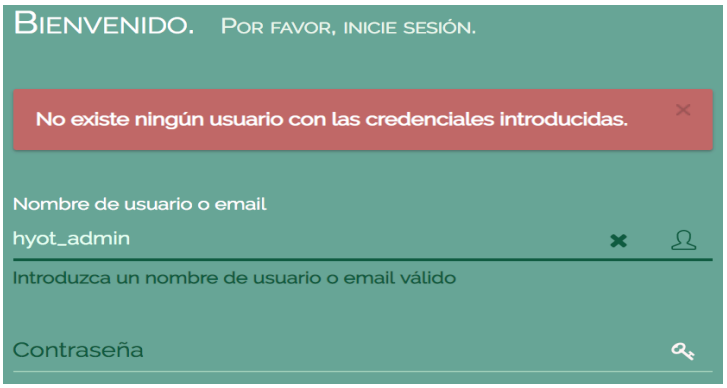


Respecto a la funcionalidad para iniciar sesión se pueden presentar los siguientes casos:

- Inicio de sesión correcto: el usuario introduce correctamente sus credenciales y el sistema le permite el acceso.
- Inicio de sesión incorrecto (Figura H.83): el usuario introduce erróneamente las credenciales

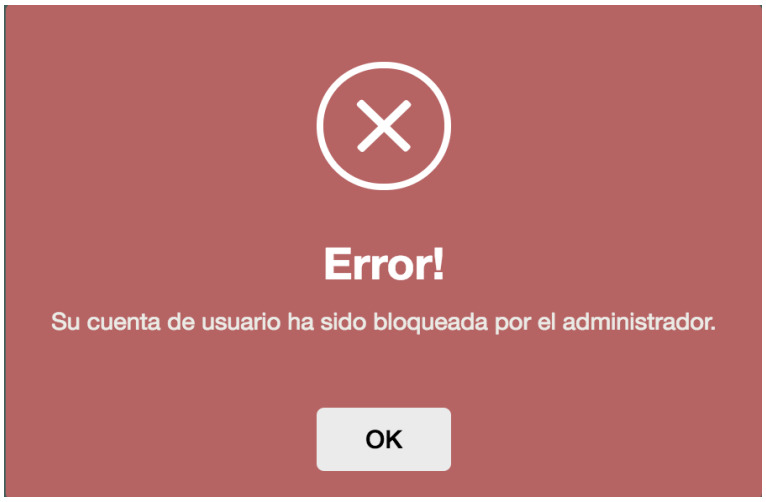
y el sistema muestra una notificación[18] al usuario, denegándole el acceso.

Figura H.83: Sistema web - Iniciar sesión con credenciales erróneas.



- Error durante la autenticación: el sistema experimenta un error durante el proceso de autenticación. En este caso, se recomienda intentar de nuevo el inicio de sesión transcurrido un tiempo.
- Cuenta de usuario no habilitada o bloqueada: la cuenta ha sido bloqueada o inhabilitada por el usuario administrador. En este caso, se muestra una alerta informativa (Figura H.84).

Figura H.84: Sistema web - Cuenta de usuario bloqueada.



- Cuenta de usuario o contraseña expirada: la cuenta o contraseña se encuentra expirada debido a que un usuario administrador así lo indicó en el sistema. Se proporciona la misma

[18]Todas las notificaciones que se presentan al usuario, excepto aquellas que requieren de su interacción, poseen un temporizador de 5 segundos tras el cual dejan de ser visibles.

funcionalidad que en el anterior caso.

Tras una autenticación exitosa por parte del usuario el sistema redirecciona a éste en base a su rol, existiendo dos roles:

- Rol usuario administrador: aquel usuario encargado de gestionar los usuarios del sistema y de visualizar la información completa monitorizada. Accede al panel de control.
- Rol usuario normal: aquel usuario que utiliza el sistema para visualizar su propia información monitorizada. Accede a la página de usuario.

Además, existe la opción *Recordarme* la cual permite ofrecer al usuario la posibilidad de recordar las credenciales para próximos inicios de sesión. De esta forma, el usuario al acceder al sistema no deberá introducir las credenciales y se producirá una autenticación automática. Es importante no utilizar esta opción en lugares públicos o en dispositivos ajenos. En caso contrario, es importante cerciorarse de finalizar la sesión.

Por último, aunque no se presenta durante un funcionamiento correcto, existe un hipotético caso donde un usuario puede que no posea un rol asignado debido a algún fallo interno durante su creación. En este caso el usuario no dispone del privilegio de acceso a las funcionalidades del sistema y por ello cuando inicia sesión es redirigido a una página informativa (Figura H.85).

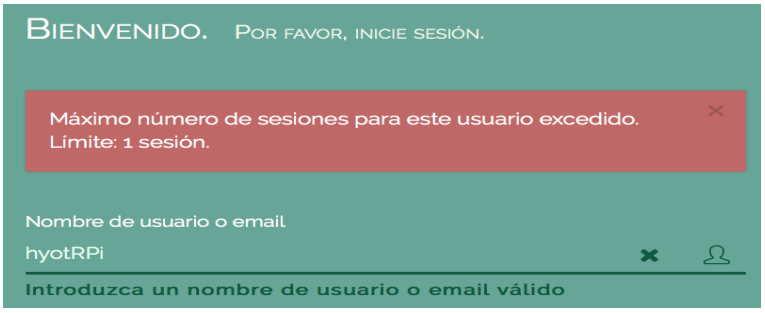
Figura H.85: Sistema web - Cuenta de usuario sin rol.

H.4.2. Sesiones concurrentes

El concepto de sesiones concurrentes indica el número de veces que un usuario puede iniciar sesión simultáneamente en el sistema desde diferentes lugares, dispositivos y/o navegadores. La política establecida es la siguiente:

- Las cuentas de usuario administrador no poseen ninguna restricción, pudiendo iniciar tantas sesiones simultáneas como deseen con su cuenta de usuario.
- Las cuentas de usuario normal y cuentas de usuario sin rol solamente pueden iniciar una sesión simultánea, de tal forma que si hay una sesión ya iniciada con un determinado usuario y se intenta iniciar sesión desde otro lugar, dispositivo y/o navegador con el mismo usuario, el inicio de sesión es denegado y se muestra una notificación de error (Figura H.86).

Figura H.86: Sistema web - Sesiones concurrentes con usuario normal



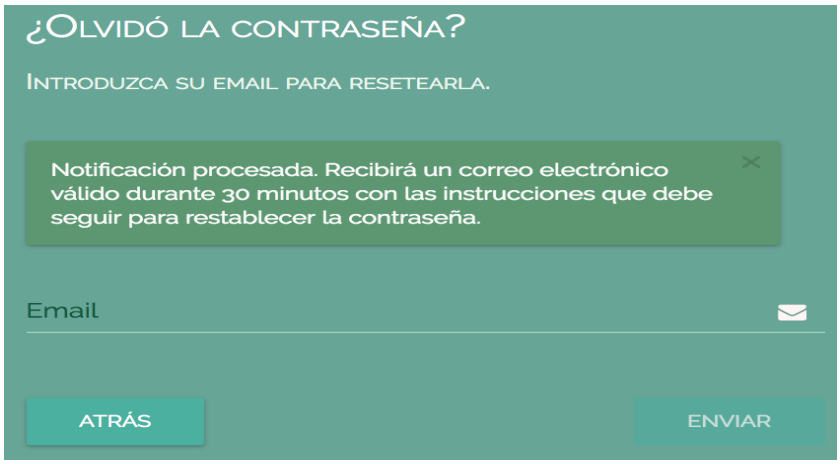
H.4.3. Restablecer contraseña

Si el usuario no recuerda su contraseña puede utilizar esta funcionalidad para su restablecimiento (Figura H.87). Esta página puede ser accedida pulsando en el enlace ubicado en la página inicial de inicio de sesión y basta con introducir el *email* registrado en el sistema y una notificación avisando del envío de un correo electrónico es obtenida[19] (Figura H.88). En el caso de que se produzca un fallo interno durante el envío del *email*, una notificación de error es mostrada.

[19]Únicamente se envía el correo electrónico si la dirección introducida existe en el sistema. En caso de no existir un usuario con dicha dirección también se muestra la notificación de envío de correo electrónico aunque internamente no se envíe con el fin de evitar la enumeración de usuarios.

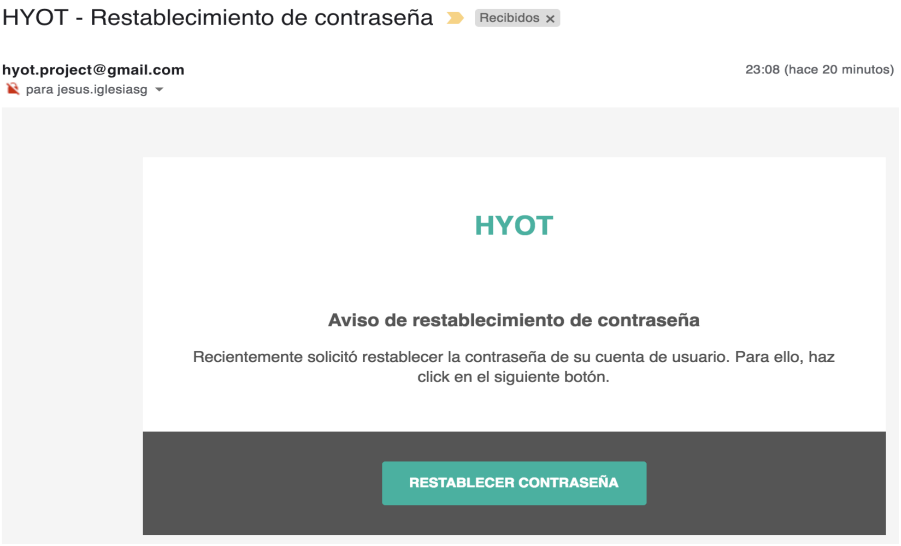
Figura H.87: Sistema web - Restablecer contraseña.

Figura H.88: Sistema web - Notificación de restablecimiento de contraseña.



La Figura H.89 muestra un ejemplo de formato del correo electrónico que se envía el cual contiene las instrucciones a seguir.

Figura H.89: Sistema web - Email para restablecer contraseña.



En la página que se redirecciona desde el correo electrónico enviado (Figura H.90), el usuario puede introducir la nueva contraseña siguiendo el patrón permitido. Si la contraseña no cumple los requisitos, una notificación de error es mostrada. Esta funcionalidad de un solo uso en cada petición solamente se encuentra disponible durante los 30 primeros minutos desde la recepción del *email* por parte del usuario, donde una vez transcurrido ese tiempo la petición actual ya no es válida, teniendo que volver a solicitar la petición de restablecimiento de contraseña.

Figura H.90: Sistema web - Indicación de nueva contraseña.

H.4.4. Panel de control

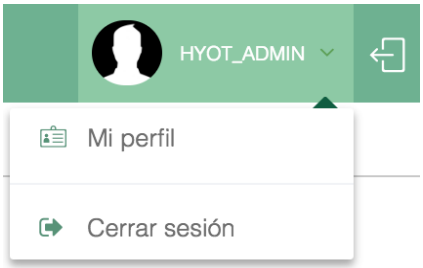
El panel de control corresponde a la parte del sistema web donde únicamente puede acceder el usuario administrador con el fin de gestionar los usuarios y consultar la información completa de monitorización por lo que ninguna acción que se pueda realizar en este panel puede ser efectuada por un usuario normal o un usuario no registrado. En el panel de control se localizan dos tipo de menú, a continuación detallados.

- Menú superior (Figura H.91), donde se encuentra:

Figura H.91: Sistema web - Menú superior del panel de control.

  - Logo de Hyot: pulsando sobre él, el usuario es redirigido a la página inicial del panel de control la cual corresponde con el panel de estadísticas para el usuario administrador.
  - Icono para ocultar/visualizar el menú lateral.
  - Imagen de perfil del usuario administrador conectado. Si el usuario no establece una imagen, se muestra una por defecto como muestra la figura anterior.
  - Nombre de usuario del usuario administrador conectado.
  - Lista desplegable que contiene las siguientes opciones (Figura H.92):

Figura H.92: Sistema web - Desplegable en el menú superior del panel de control.

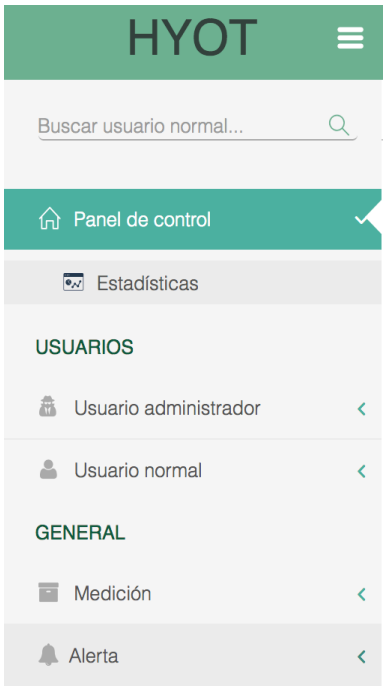


    - Mi perfil: permite visualizar el perfil personal del usuario.
    - Cerrar sesión: permite cerrar la sesión del usuario invalidando también el inicio automático en el actual dispositivo si estuviese habilitado.
  - Cerrar sesión: misma funcionalidad que la opción situada en la lista desplegable.

- Menú lateral (Figura H.93), donde se encuentra:

- Búsqueda rápida de usuarios normales.
- Panel de control → Estadísticas: permite visualizar la información y gráficos estadísticos sobre el proyecto Hyot.
- Gestión de cada entidad: permite la gestión completa de los usuarios, usuario administrador y usuario normal.
- Visualización de las mediciones tomadas por todos los usuarios.
- Visualización de las alertas registradas en la BC por todos los usuarios.

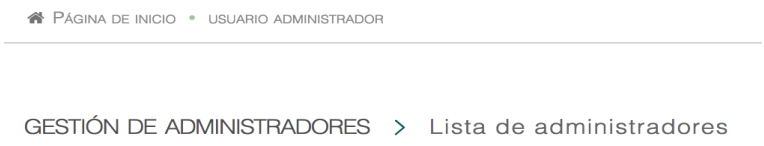
Figura H.93: Sistema web - Menú lateral del panel de control.



También, se debe mencionar que todas las páginas que se muestran en el sistema web presentan una estructura común, formada por (Figura H.94):

- Guía de navegación: permite en todo momento al usuario administrador conocer la ubicación actual dentro del sistema web. Pulsando sobre el texto *Página de inicio*, el usuario administrador es redirigido a la página inicial del panel de control.
- Título y subtítulo de la sección: permite conocer al usuario en qué sección y subsección se encuentra actualmente. Al pulsar sobre el texto que indica la sección, el usuario administrador es redirigido a la acción principal de esa entidad, en el caso de la figura al listado de usuarios administradores.

Figura H.94: Estructura - Guía de navegación, título y subtítulo.



Panel de estadísticas

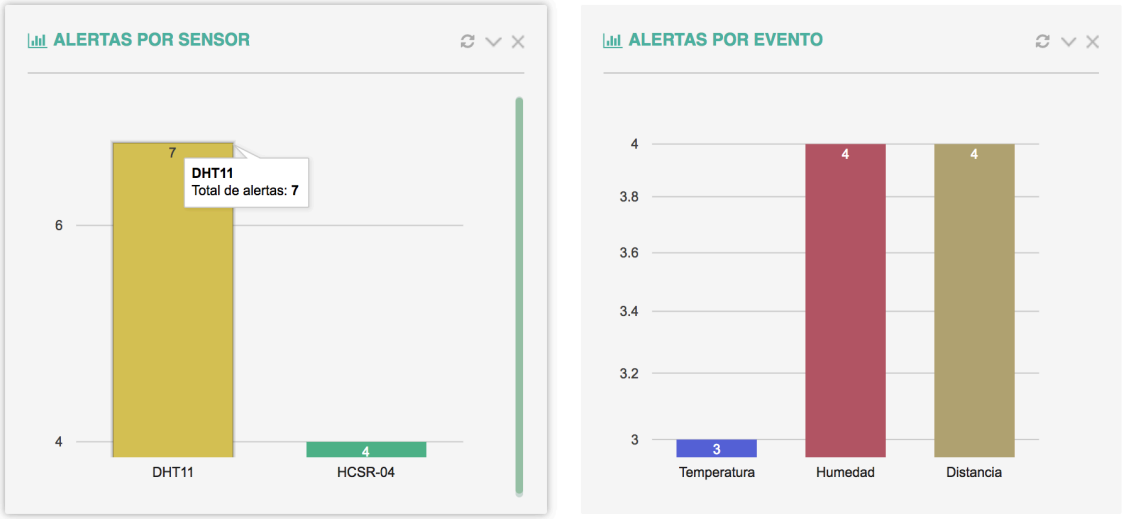
El panel de estadísticas permite al usuario administrador conocer más información sobre determinados datos estadísticos los cuales son recopilados y mostrados en formato numérico o gráfico con el fin de facilitar la interpretación y análisis sobre ellos. A continuación, se detallan los diferentes datos y gráficos mostrados:

- Datos estadísticos (Figura H.95), donde se refleja:
  - Número de usuarios administradores registrados en el sistema web.
  - Número de usuarios normales registrados en el sistema web.
  - Número de mediciones sobre el entorno almacenadas por todos los usuarios.
  - Número de alertas registradas en la BC por todos los usuarios.
  - Número de usuarios registrados en la BC.

Figura H.95: Sistema web - Datos estadísticos.

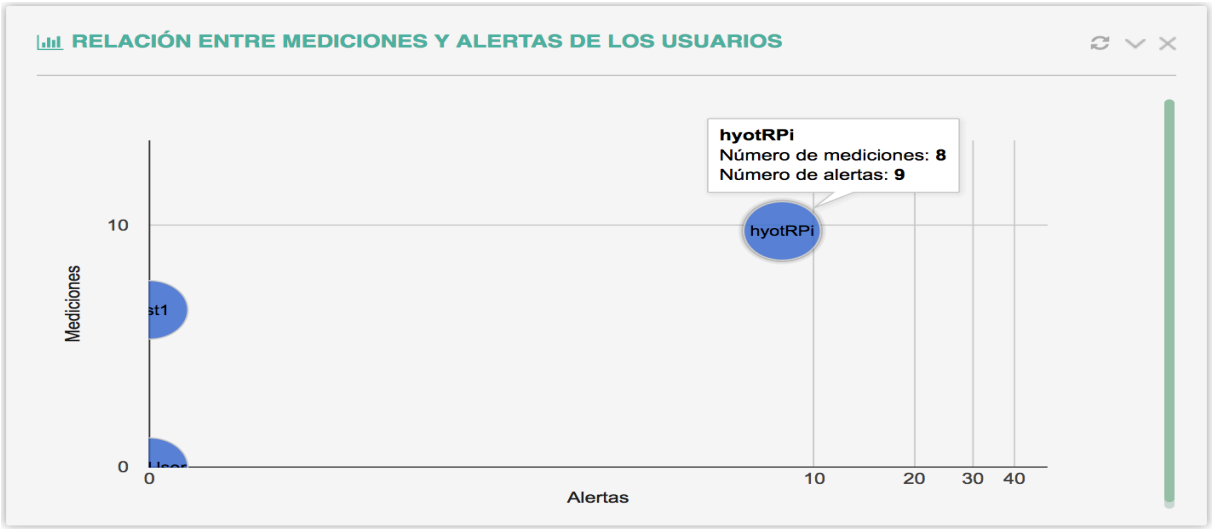
- Alertas registradas en la BC por cada sensor y cada evento (Figura H.96).

Figura H.96: Sistema web - Alertas registradas por cada sensor y evento.



- Relación de medidas tomadas y alertas registradas en la BC por cada usuario existente en este método de persistencia (Figura H.97).

Figura H.97: Sistema web - Relación de medidas y alertas por cada usuario.



- Usuarios recientes (Figura H.98): este panel muestra los 10 últimos usuarios normales dados de alta en el sistema web junto con información básica: nombre de usuario, email, fecha de alta y estado de la cuenta. Además, el botón *Editar* permite acceder a la página de edición de este usuario de forma rápida.

Figura H.98: Sistema web - Estadísticas sobre usuarios normales recientes.

Cada sección del panel estadístico puede contener una serie de iconos (Figura H.99) con una determinada funcionalidad[20].

- Recargar: permite recargar de forma individual los datos del gráfico o sección seleccionada de una manera rápida sin tener que recargar la página completa.
- Agrupar/Expandir: permite agrupar o expandir la sección seleccionada.
- Remover: permite ocultar la sección seleccionada. Para visualizarla de nuevo basta con recargar la página completa.

Figura H.99: Sistema web - Funcionalidad de iconos del panel estadístico.

Gestión de usuarios administradores

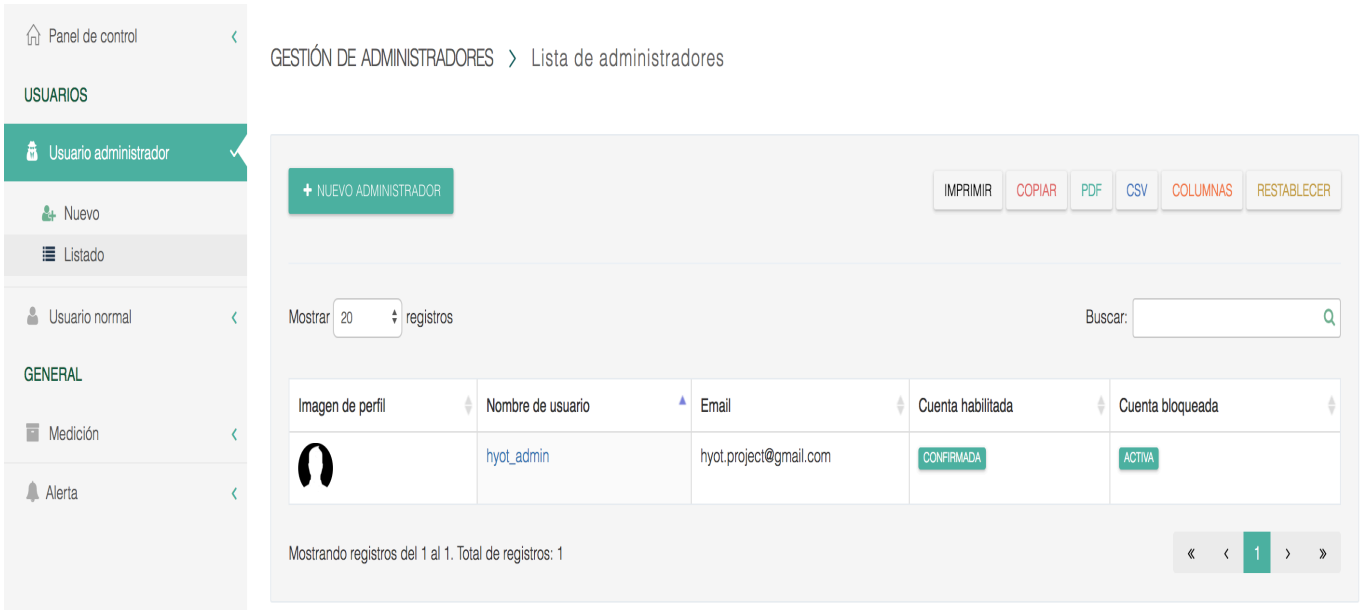
La gestión de usuarios administradores permite las actividades[21] principales de: listado, creación, edición, eliminación y exportación -subactividad de listado-. Las acciones de creación y listado de usuarios administradores pueden ser accedidas directamente desde el menú lateral, visualizándose en color verde el título de la entidad de usuario administrador cuando es seleccionada alguna acción. A continuación, se explica en detalle cada acción disponible sobre la gestión de usuarios administradores:

- Listado de usuarios administradores (Figura H.100): permite consultar los usuarios administradores existentes en el sistema web. En esta sección se encuentra:

[20]Se muestran todos los iconos existentes juntos y se explican de izquierda a derecha aunque puede darse el caso que una sección determinada presente todos y otra sección solamente algunos.

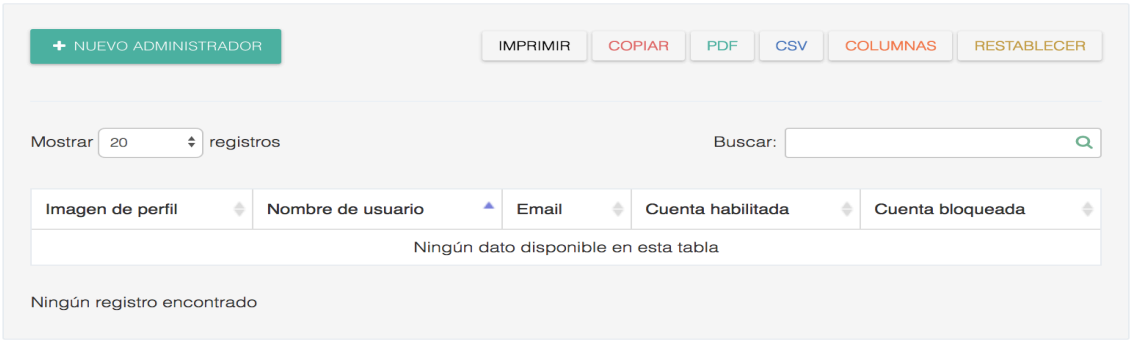
[21]Las acciones explicadas en este apartado son similares y aplicables a la entidad usuario normal.

Figura H.100: Sistema web - Listado de usuarios administradores.



- La visualización de todos los usuarios administradores existentes en el sistema ordenados y paginados. Por defecto solamente los campos del usuario administrador marcados como principales son visualizados a través de las columnas. El resto de campos, para el caso actual *Cuenta expirada* y *Contraseña expirada* se encuentran ocultos y pueden ser visualizados en cualquier momento a través de la selección de las columnas correspondientes. En el caso de que no exista ningún usuario administrador en el sistema web o la búsqueda no proporcione ningún resultado coincidente, se muestra un mensaje informativo (Figura H.101).

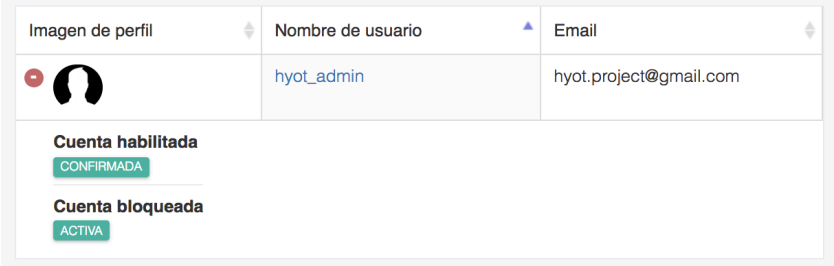
Figura H.101: Sistema web - Listado vacío de usuarios administradores.



Además, si la visualización del sistema web se realiza en un dispositivo con un tamaño de pantalla menor, la tabla que muestra el listado de administradores se ajusta a las

dimensiones actuales no mostrando por tanto todos los campos visibles. En dicho caso, se muestra un icono que permite extender y replegar la fila para ver el resto de columnas (Figura H.102).

Figura H.102: Sistema web - Listado adaptativo de usuarios administradores.



- La posibilidad de búsqueda de usuarios administradores por cualquier campo existente.
- La posibilidad de ordenar de manera ascendente o descendente los registros según una determinada columna. Basta con pulsar en el título de la columna para la ordenación, siendo la flecha morada la que indique el sentido de ordenación.
- La posibilidad de filtrar el número de registros a mostrar (por defecto, 20). En caso de existir más registros que la cantidad filtrada, la paginación -barra de navegación situada en la sección inferior- se activa para poder visualizar las diferentes páginas de usuarios administradores.
- La posibilidad de modificar el orden de las columnas pulsando sobre una y arrastrándola hacia otra posición.
- La posibilidad de aplicar las siguientes opciones situadas en la barra de opciones:
  - Imprimir: imprime todos[22] los usuarios administradores.
  - Copiar: copia al portapapeles los usuarios administradores.
  - Exportar a PDF: exporta los usuarios administradores a un fichero PDF.
  - Exportar a CSV: exporta los usuarios administradores a un fichero CSV.
  - Columnas: permite indicar qué columnas visualizar en ese instante.
  - Restaurar: permite volver a mostrar las columnas establecidas como principales.
- La posibilidad de crear un nuevo usuario administrador.
- La edición o eliminación de un usuario determinado pulsando sobre el registro deseado de la columna que contiene el nombre de usuario.
- La adicción de una imagen de perfil la cual puede ser introducida también durante el proceso de creación.

[22]En el caso de realizar un búsqueda, solamente se imprimen aquellos usuarios administradores resultantes tras la búsqueda. Esto también aplica a las acciones de copiar, exportar a PDF y CSV.

- Creación de un usuario administrador: permite crear un nuevo usuario administrador tras rellenar correctamente los datos solicitados donde algunos de ellos son obligatorios -marcados con un asterisco en rojo- y otros opcionales. En esta pantalla se pueden diferenciar dos secciones:

  - Información de la cuenta de usuario (Figura H.103), donde todos los campos son requeridos.
    - Nombre de usuario: nombre utilizado para iniciar sesión. Debe ser único en el sistema web[23] y por ello se ofrece la verificación de disponibilidad.
    - *Email*: dirección electrónica utilizada para iniciar sesión y como medio de contacto. Debe ser única en el sistema web.
    - Contraseña: credencial de acceso al sistema web. Debe cumplir un patrón específico y a medida que se introduce -nunca es mostrada en texto legible- aparece en la parte inferior una barra indicando su fortaleza.
    - Confirmar contraseña: mecanismo para asegurar que la contraseña introducida es la deseada. Debe coincidir con el campo *Contraseña*.
    - Cuenta expirada: permite expirar la cuenta del usuario administrador (por defecto, deshabilitada). Si esta opción se habilita el usuario no puede acceder al sistema.
    - Cuenta bloqueada: permite bloquear la cuenta del usuario administrador (por defecto, deshabilitada). Si esta opción se habilita el usuario no puede acceder al sistema. Se recomienda utilizar esta funcionalidad para restringir el acceso a los usuarios en caso de ser necesario[24].
    - Cuenta habilitada: permite activar la cuenta del usuario administrador (por defecto, habilitada). Si esta opción se deshabilita el usuario no puede acceder al sistema.
    - Contraseña expirada: permite expirar la contraseña de la cuenta del usuario administrador (por defecto, deshabilitada). Si esta opción se habilita el usuario no puede acceder al sistema.
  - Imagen de perfil (Figura H.104): permite seleccionar una imagen para el perfil del usuario. En caso de no establecer ninguna, se muestra una imagen por defecto previamente establecida. Esta imagen debe cumplir una serie de requisitos: ser de tipo PNG, JPEG o GIF y su tamaño no puede superar 1 MB.

[23] Todos los campos que debe ser únicos presentan la funcionalidad de comprobación de disponibilidad in situ.

[24] El orden de prioridad de los estados de la cuenta de un usuario es: cuenta bloqueada, cuenta habilitada, cuenta expirada y contraseña expirada.

Figura H.103: Sistema web - Crear nuevo usuario administrador.

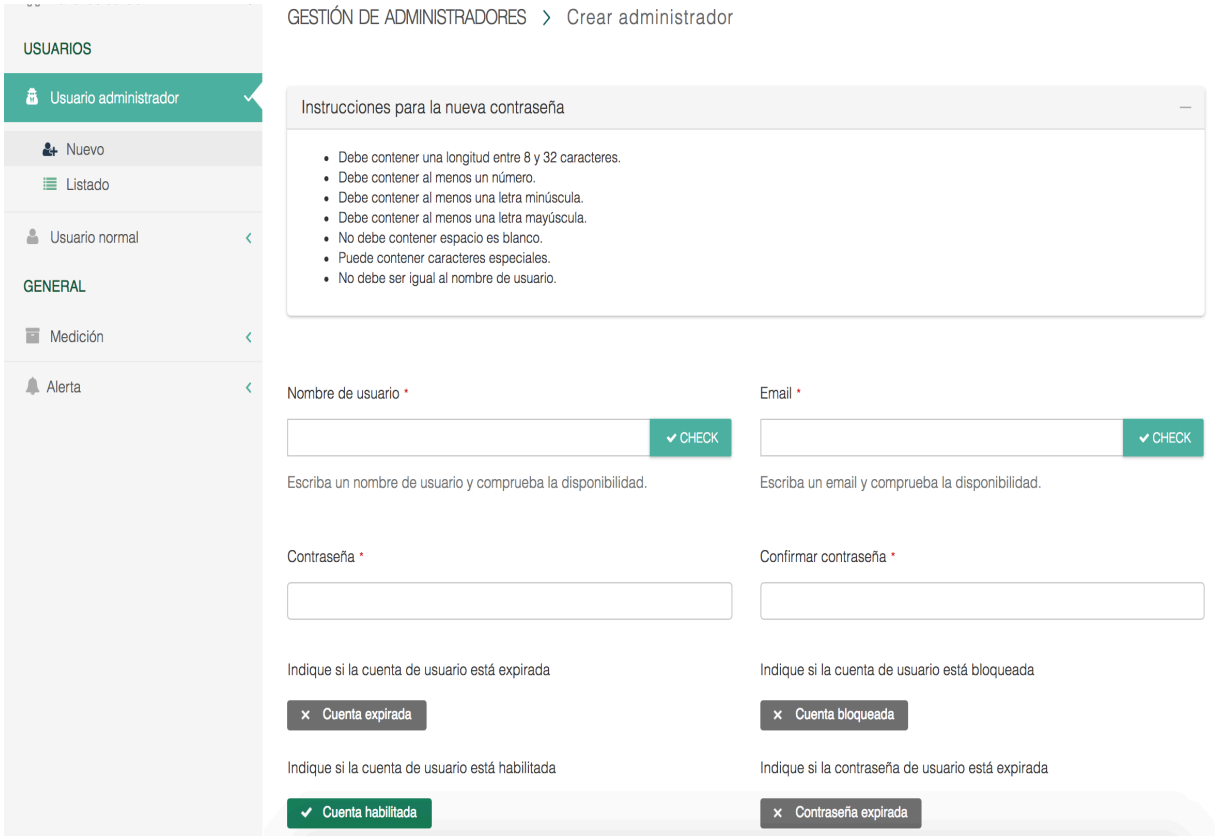
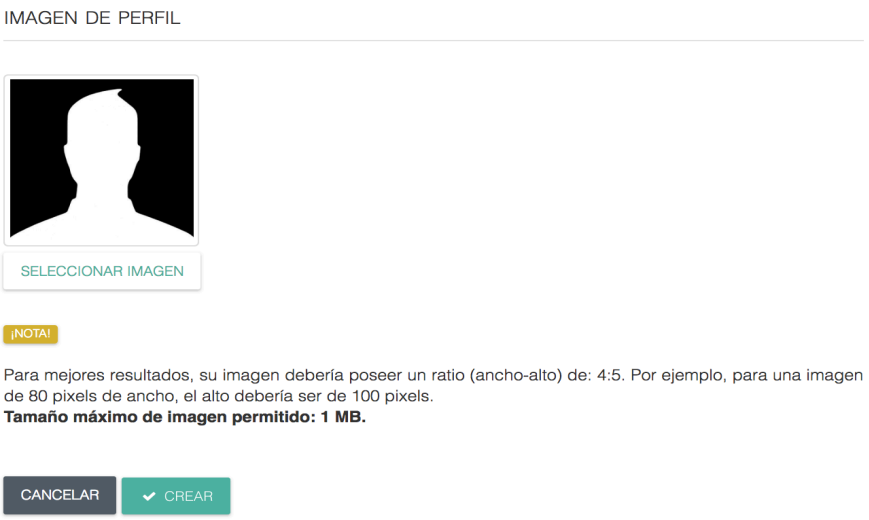


Figura H.104: Sistema web - Seleccionar imagen de perfil durante la creación de un nuevo usuario administrador.



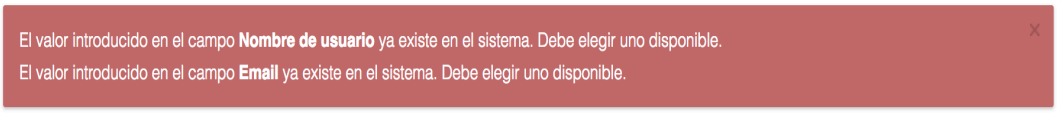
Durante la introducción de los datos, éstos son verificados[25] con respecto a una serie de reglas para informar al usuario sobre si son válidos -aparece un icono verde- o inválidos -aparece un icono rojo. En este último caso se muestra un mensaje informativo, tal y como muestra la Figura H.105.

Figura H.105: Sistema web - Validación de campos.

De la anterior figura también se puede observar la aparición de un icono aspa[26] (×) gris situado a la derecha del campo de texto cuando se introducen datos. En el caso de tratarse de un campo normal, el icono permite borrar el texto del campo. Por contra si se trata de un campo contraseña, el icono permite visualizar la contraseña en texto legible.

Además existen otras situaciones donde la validación de los campos se produce una vez confirmada la acción de creación. En este caso si existe algún error se muestra mediante una notificación (Figura H.106).

Figura H.106: Sistema web - Notificación de datos erróneos.



- Edición de un usuario administrador: permite editar el usuario administrador seleccionado. En esta pantalla aparecen todos los campos del usuario administrador con su valor actual[27], excepto la imagen de perfil la cual se edita desde su propio apartado. Para confirmar la edición el usuario debe confirmar la actualización de la información y una notificación de éxito o error se muestra al usuario. A la hora de actualizar se realiza una comprobación

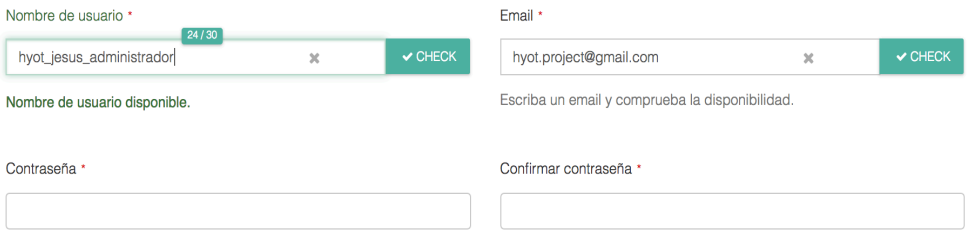
[25] Las validaciones y notificaciones se llevan a cabo durante todas las acciones de una entidad: creación, edición y eliminación.

[26] Estos iconos cuya funcionalidad es facilitar la experiencia de usuario aparecen en casi todos los campos de texto del sistema web.

[27] Los campos con contenido sensible como son las contraseñas nunca son reveladas en texto legible.

de edición concurrente, es decir, si hay otro usuario administrador que está editando la misma instancia de forma simultánea. En este caso no se realiza la actualización para evitar inconsistencias y se muestra una notificación de error. La Figura H.107 muestra un ejemplo de edición del nombre de usuario donde observa cómo aparece una ayuda visual sobre el número de caracteres permitidos en el campo.

Figura H.107: Sistema web - Editar usuario administrador.



- Eliminación de un usuario administrador (Figura H.108). Para eliminar un usuario administrador tan solo se debe seleccionar el usuario deseado desde el listado y en la parte superior de la pantalla de edición aparece el botón *Eliminar administrador*. Cuando este botón se pulsa, aparece una ventana de confirmación, donde:
  - Eliminar: confirma el borrado del usuario administrador mostrando posteriormente una notificación al usuario.
  - Cancelar: cancela la eliminación del usuario administrador. También se cancela la eliminación si se pulsa en cualquier lugar de la pantalla.

Figura H.108: Sistema web - Eliminar usuario administrador.

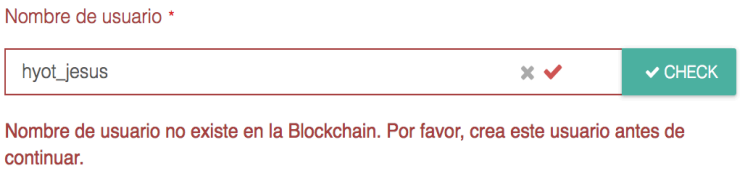
Gestión de usuarios normales

Los usuarios normales representan a los usuarios que consumen la información propia en tiempo real sobre la monitorización del entorno IoT, es decir, aquella información -mediciones y alertas-

que ha sido registrada por un usuario especificado en el componente de monitorización de sucesos con el mismo nombre de usuario. Un usuario normal posee los mismos campos que el usuario administrador donde a mayores se incluyen dos campos de información obligatorios: nombre y apellidos.

Desde esta sección, el usuario administrador puede gestionar completamente los usuarios normales del sistema web de una forma similar a la gestión del usuario administrador. Únicamente existe una diferencia durante la creación del usuario normal. Un usuario normal debe estar relacionado por el nombre de usuario con un usuario previamente existente en la BC, de forma que en el sistema web únicamente existen usuarios normales que son usuarios que monitorizan el entorno. Esto implica que a la hora de su creación, el nombre de usuario elegido deba asociarse al nombre de usuario de un usuario de la BC y deba estar disponible en el sistema web. En caso contrario, el usuario normal no puede ser creado en el sistema web, tal y como muestra la Figura H.109.

Figura H.109: Sistema web - Comprobación de nombre de usuario de un usuario normal.



En el caso de que el servidor de HC no se encuentre en ejecución, la petición de comprobación de usuario no puede ser efectuada por lo que el usuario no puede ser creado y se muestra la notificación de la Figura H.110.

Figura H.110: Sistema web - Servidor Blockchain no se encuentra en ejecución durante la comprobación.

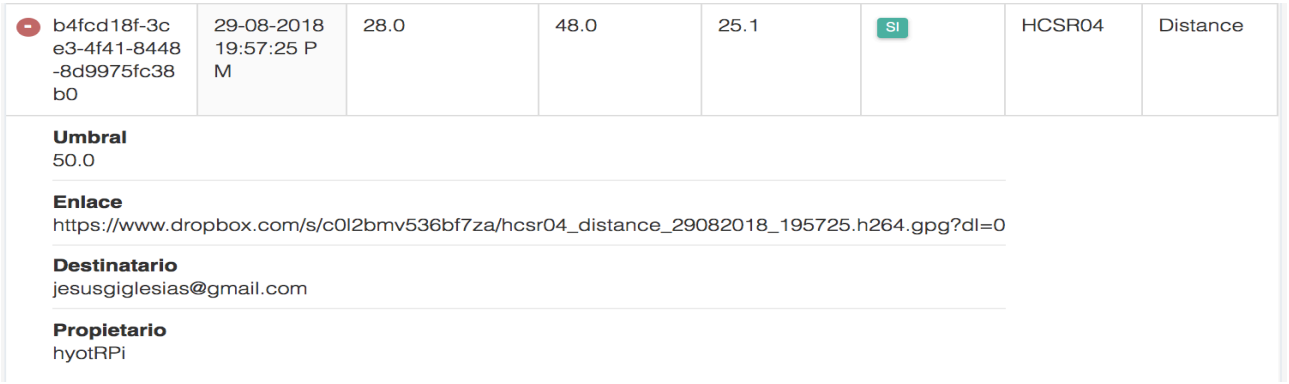
Visualización de mediciones monitorizadas

En esta sección el usuario administrador puede visualizar todas las mediciones almacenadas en la BBDD por los distintos usuarios. Al igual que en la acción de listar de las entidades anteriores, aquí también se presentan todas las funcionalidades comentadas anteriormente: buscar, ordenar,

filtrar número de registros, copiar, imprimir, exportar, etc. La Figura H.111 muestra el listado donde se observan 3 mediciones del entorno activándose el protocolo de alerta en una de ellas. Además de los campos visibles, existen otros dos: *Enlace* y *Destinatario* los cuales pueden ser visualizables habilitando la columna correspondiente tal y como muestra la Figura H.112.

Figura H.111: Sistema web - Listado de mediciones.

Figura H.112: Sistema web - Listado de todos los campos de la medición.



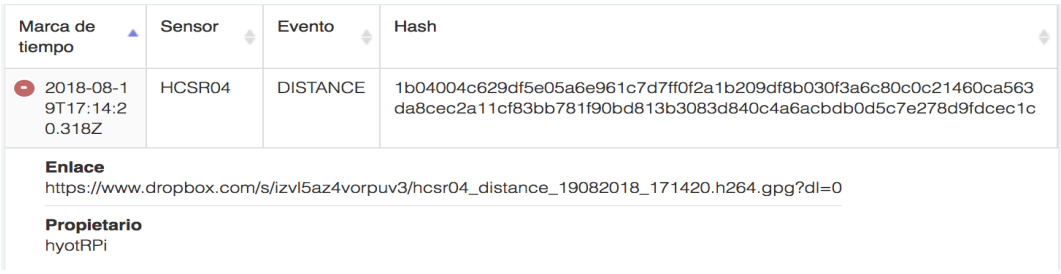
Visualización de alertas registradas

En esta sección el usuario administrador puede visualizar todas las alertas almacenadas en la BC por los distintos usuarios. El listado de alertas (Figura H.113) contiene las mismas funciona-

lidades que las ya comentadas y únicamente el campo *Hash* está oculto por defecto. El valor de este campo (Figura H.114) es el utilizado por el usuario cuando quiera verificar la originalidad de la evidencia en el componente de desencriptación de evidencia.

Figura H.113: Sistema web - Listado de alertas.

Figura H.114: Sistema web - Listado de todos los campos de la alerta.



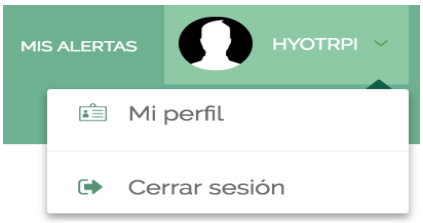
H.4.5. Página de usuario

La página de usuario corresponde a la parte del sistema web donde únicamente puede acceder el usuario normal con el fin de consumir la información monitorizada de la que sea poseedor. Cada usuario normal registrado en el sistema web está asociado con un usuario en la BC con el mismo nombre de usuario e incluso pudiendo coincidir el *email*, el nombre y los apellidos. Dentro de la página de usuario, única e independiente para cada uno, se encuentra una barra de menú (Figura H.115) en la parte superior con las siguientes funcionalidades:

Figura H.115: Sistema web - Menú superior del panel de control.

- Logo de Hyot: pulsando sobre él, el usuario es redirigido a la página inicial de la página de usuario la cual corresponde con la sección estadística del propio usuario normal.
- Mis mediciones: contiene el listado de mediciones de las que el usuario normal actual es poseedor.
- Mis alertas: contiene el listado de alertas de las que el usuario normal actual es poseedor.
- Imagen de perfil del usuario normal conectado. Si el usuario no establece una imagen, se muestra una por defecto como muestra la figura anterior.
- Nombre de usuario del usuario normal conectado.
- Lista desplegable que contiene las siguientes opciones (Figura H.116):

Figura H.116: Sistema web - Desplegable en el menú superior del panel de control.



  - Mi perfil: permite visualizar y/o editar el perfil personal del usuario.
  - Cerrar sesión: permite cerrar la sesión del usuario invalidando también el inicio automático en el actual dispositivo si estuviese habilitado.
- Cerrar sesión: misma funcionalidad que la opción situada en la lista desplegable.

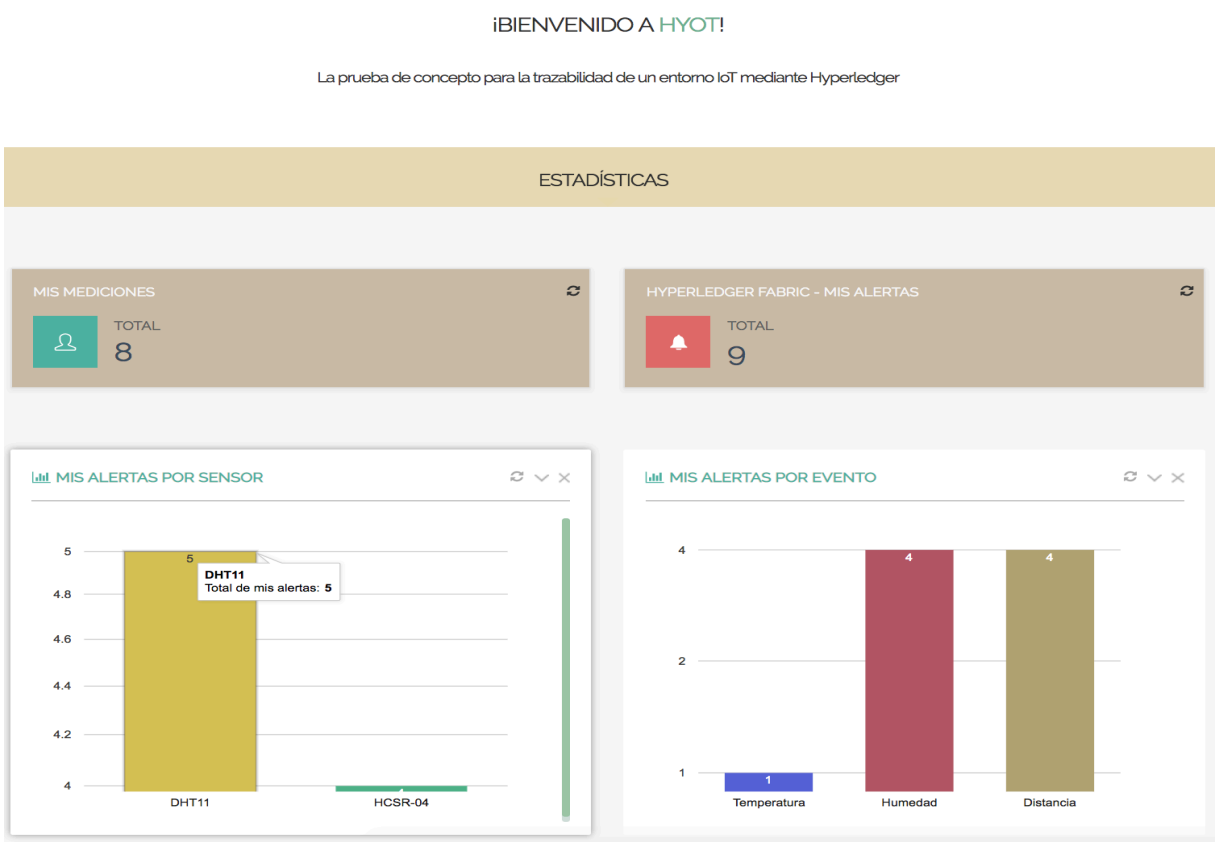
Además en la parte inferior del menú superior se presenta en todas las páginas una guía de navegación. Esta guía permite en todo momento al usuario conocer la ubicación exacta donde se encuentra. Pulsando sobre el texto: *Página de inicio*, se redirige a la página inicial de la página de usuario.

Sección estadística

La página inicial de la página de usuario contiene una sección con diferentes estadísticas sobre la monitorización llevada a cabo por el usuario normal actual (Figura H.117), donde se encuentra:

- Número de mediciones propias sobre el entorno almacenadas.
- Número de alertas propias registradas en la BC.
- Alertas propias registradas en la BC por cada sensor y cada evento.

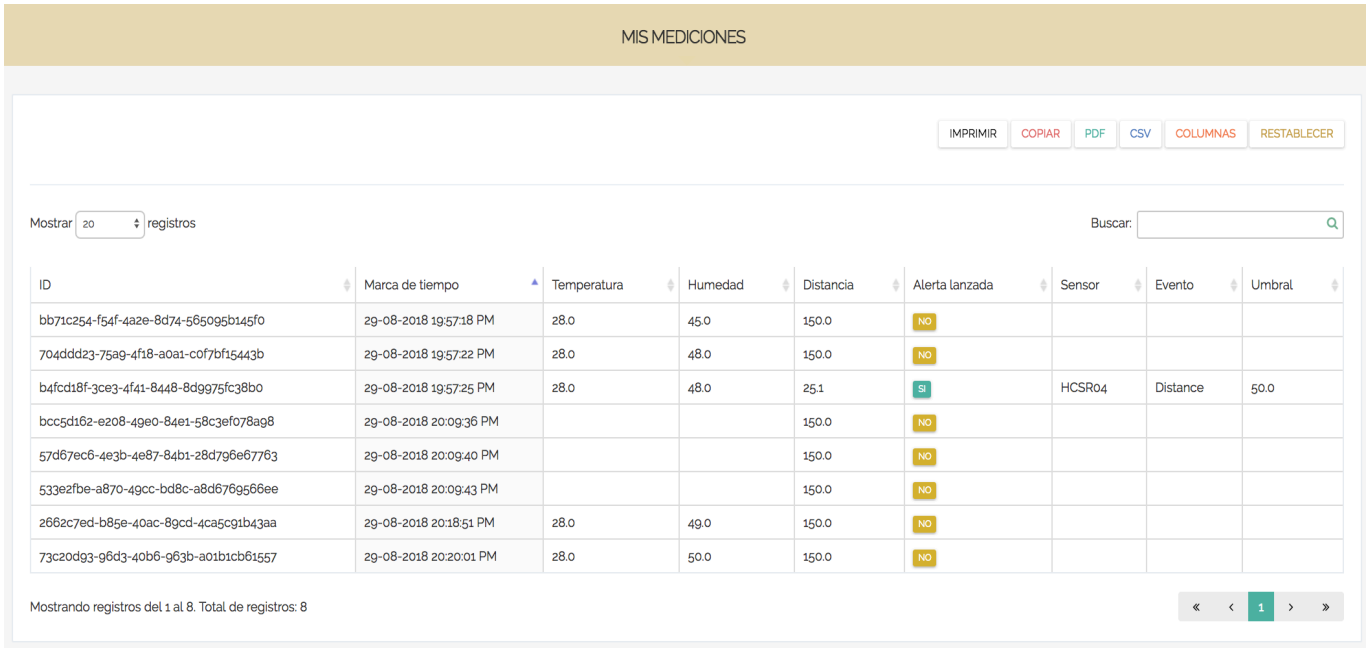
Figura H.117: Sistema web - Sección estadística del usuario normal.



Visualización de mis mediciones

En esta sección el usuario normal puede visualizar todas las mediciones almacenadas en la BBDD de las que sea poseedor. Todas las funcionalidades de la acción listar que podía realizar el usuario administrador sobre las entidades, pueden ser efectuadas también por el usuario normal sobre sus mediciones. La Figura H.118 muestra el listado de diferentes mediciones del entorno donde se observa que no todos los campos son visibles. Los campos *Enlace* y *Destinatario* pueden ser visualizables habilitando la columna correspondiente.

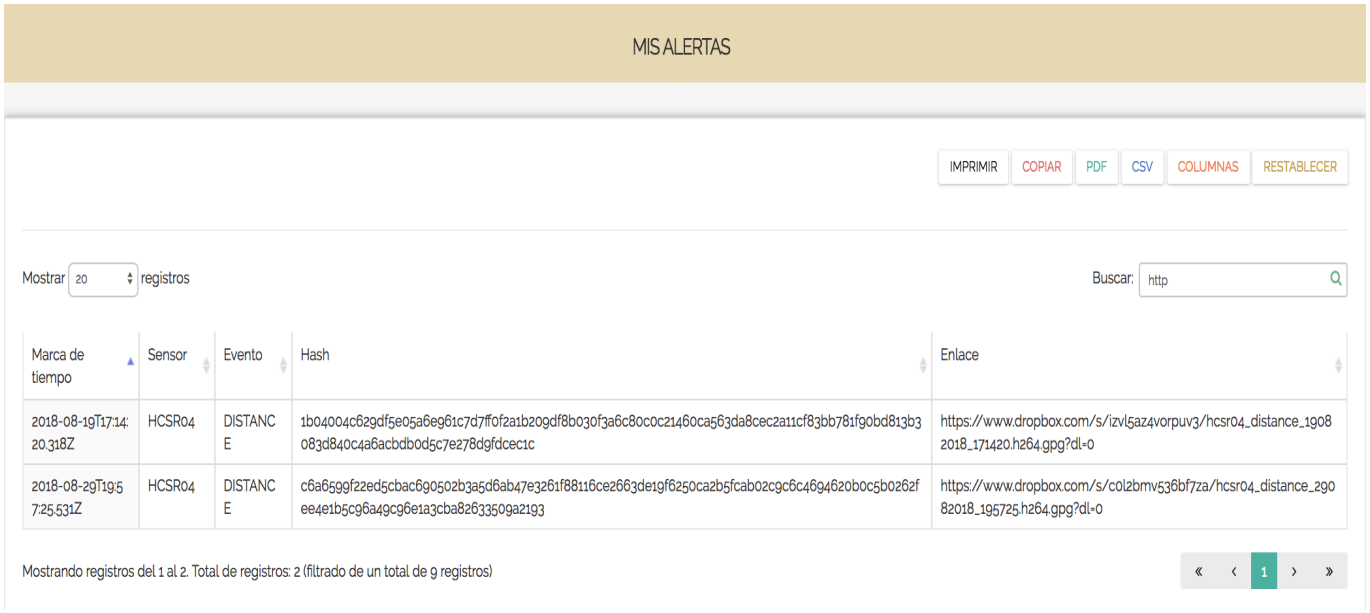
Figura H.118: Sistema web - Mediciones propias del usuario normal.



Visualización de mis alertas

En esta sección el usuario normal puede visualizar todas las alertas registradas en la BC de las que sea poseedor. La Figura H.119 muestra el listado de alertas para el usuario normal actual.

Figura H.119: Sistema web - Mediciones propias del usuario normal.



Perfil de usuario

El perfil personal del usuario normal contiene la información detallada sobre los datos personales de éste. Dentro de esta sección, el usuario normal puede visualizar y/o editar su información personal (Figura H.120), su contraseña (Figura H.121) y su imagen de perfil (Figura H.122). Cabe destacar que:

- El propio usuario no puede cambiar su nombre de usuario directamente.
- La nueva contraseña debe cumplir una serie de requisitos por razones de seguridad los cuales son indicados en la página. Además, a medida que se escribe la nueva contraseña se valida frente a una serie de reglas establecidas y se indica la fortaleza de ésta.
- La imagen de perfil debe cumplir los siguientes requisitos: ser de tipo PNG, JPEG o GIF y límite de tamaño 1 MB.

Figura H.120: Sistema web - Información personal.

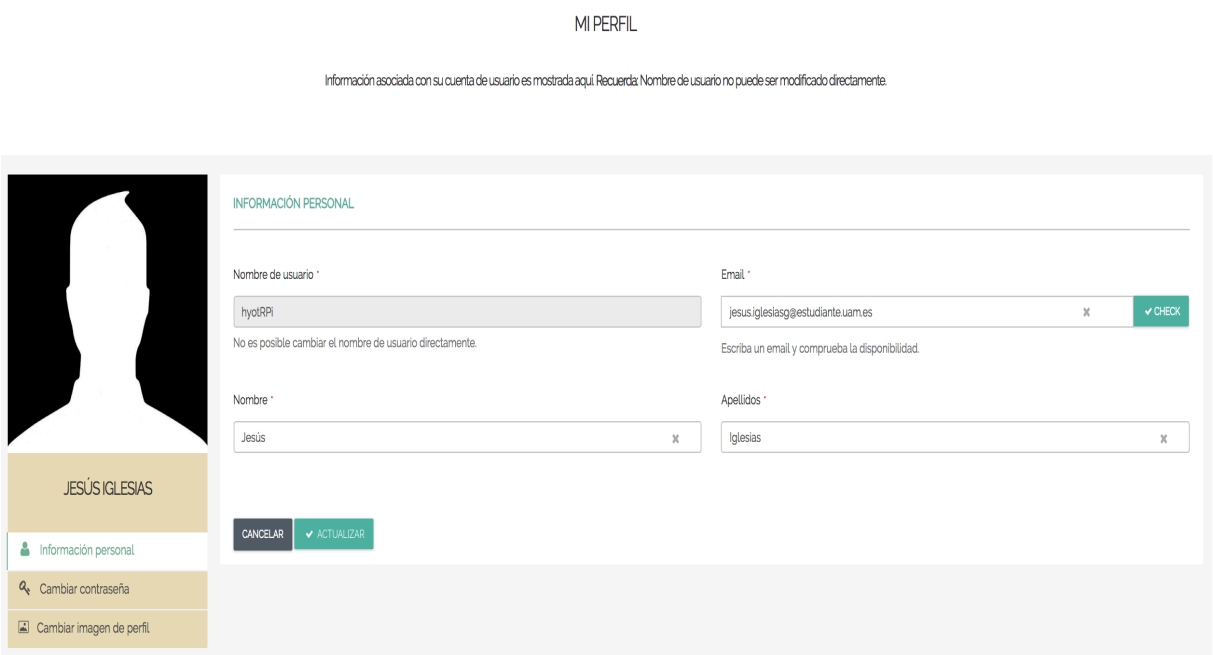


Figura H.121: Sistema web - Modificar contraseña.

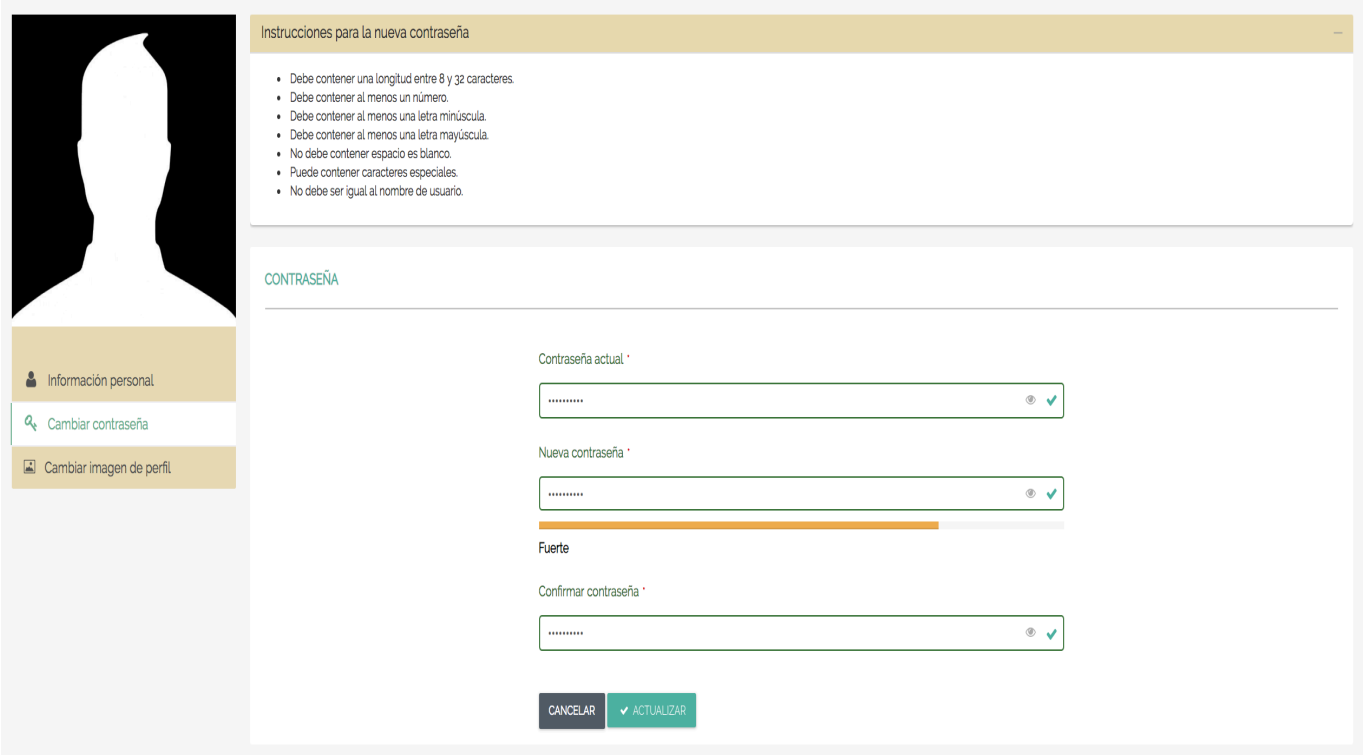
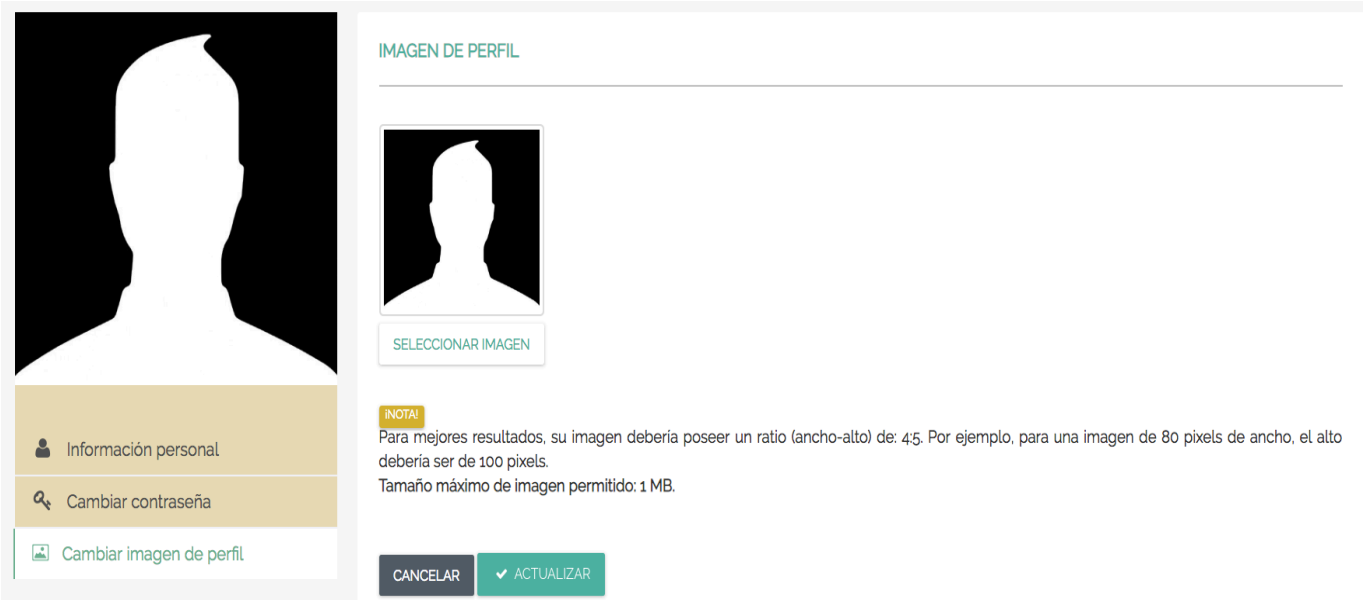


Figura H.122: Sistema web - Modificar imagen de perfil.



H.4.6. Situaciones de error

En cualquier instante, el usuario durante el uso del sistema web puede desencadenar una acción que provoque un error o incluso puede ser originado por cualquier otra causa. Es entonces cuando el usuario es redireccionado a la página de error apropiada donde se le indica el tipo de error ocurrido. En el sistema web se gestionan explícitamente los siguientes errores, cada uno de ellos con su propia página informativa:

- Error 400 - Petición errónea.
- Error 401 - Petición no autorizada.
- Error 403 - Acceso denegado/prohibido.
- Error 404 - Recurso no encontrado.
- Error 405 - Método de solicitud no soportado.
- Error 500 - Error interno del servidor.
- Error 503 - Servicio no disponible.

La Figura H.123 muestra un ejemplo de error 404 cuando el usuario intenta acceder a un recurso inexistente mientras que la Figura H.124 muestra un ejemplo de acceso a una acción no permitida.

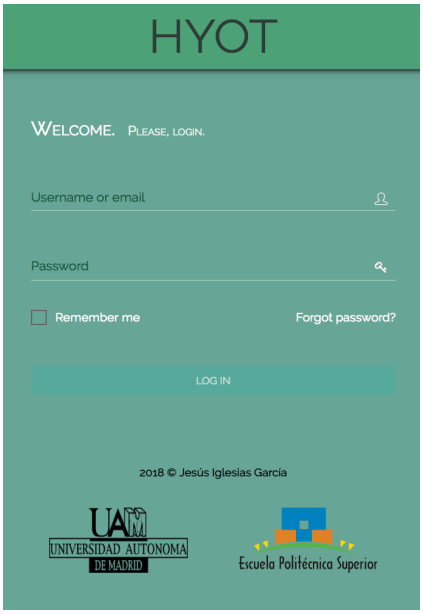
Figura H.123: Sistema web - Error 404.

Figura H.124: Sistema web - Error 403.

H.4.7. Idiomas

Actualmente el sistema web está disponible en dos idiomas, español e inglés. La imposición de uno u otro depende del idioma establecido en el propio navegador por lo que su modificación implica el cambio de idioma en el sistema web de Hyot, siempre y cuando el nuevo idioma establecido sea soportado por éste. En caso contrario, se muestra el idioma por defecto, inglés. La Figura H.125 muestra la página de inicio en el idioma inglés.

Figura H.125: Sistema web - Página de inicio en inglés.



H.4.8. Diseño adaptativo

Hyot está diseñado para una presentación óptima de la interfaz de usuario (*User Interface* -UI-) en diferentes dispositivos con diversidad de tamaños de pantallas, gracias al diseño adaptativo. En todos los casos se ofrece la misma funcionalidad que pudiera tener si su visualización se realizase en dispositivos con pantallas de mayor tamaño. La Figuras H.126 y H.127 muestran la visualización del panel de control y de la página de usuario, respectivamente desde un dispositivo con un tamaño de pantalla menor. En ambos casos, el icono situado en el extremo derecho del menú superior permite desplegar y replegar las distintas opciones de funcionalidad del sistema.

Figura H.126: Sistema web - Visualización del panel de control en un dispositivo móvil.

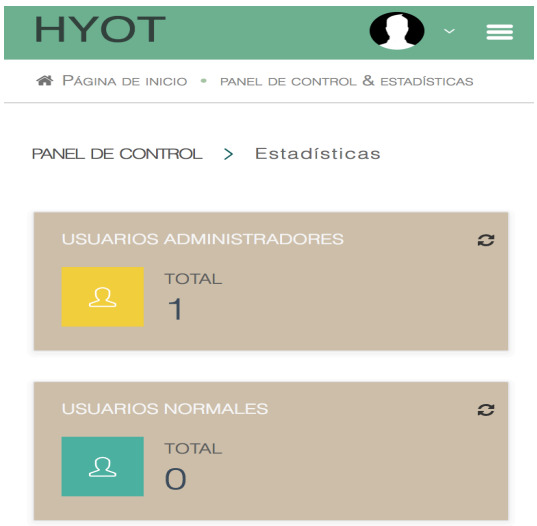


Figura H.127: Sistema web - Visualización de la página de usuario en un dispositivo móvil.